

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«Ғұмарбек Дәукеев атындағы Алматы энергетика және байланыс
университеті» коммерциялық емес АҚ
Ақпараттық технологиялар институты
IT-инжиниринг кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

PhD, доцент

_____ А.У. Утегенова

« ____ » _____ 2021 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған web-сайт әзірлеу

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

Орындаған: Мұхтария А.Қ. Тобы: ВТк-17-1

Ғылыми жетекші: доцент Мамырбаев О.Ж.

Кеңесшілер:

Экономикалық бөлім: доцент _____ Г.Ш. Боканова

« ____ » _____ 2021 ж.

Өміртіршілік қауіпсіздігі: аға оқытушы _____ Ш.А. Абдрешов

« ____ » _____ 2021 ж.

Программалық қамтама бөлімі: аға оқытушы _____ Ә.А. Айтқазина

« ____ » _____ 2021 ж.

Норма бақылаушы: аға оқытушы _____ П. Оралхан

« ____ » _____ 2021 ж.

Сын-пікір беруші: PhD, доцент _____ Ж.М. Алибиева

« ____ » _____ 2021 ж.

Алматы, 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

«Ғұмарбек Дәукеев атындағы Алматы энергетика және байланыс университеті» коммерциялық емес АҚ

Ақпараттық технологиялар институты

IT-инжиниринг кафедрасы

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

Дипломдық жобаны орындауға берілген
ТАПСЫРМА

Білім алушы Мұхтария Айгерім Қайратқызы

Жобаның тақырыбы: Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған web-сайт әзірлеу

2020 жылғы «27» қазан № 217 университет бұйрығымен бекітілген.

Аяқталған жобаны тапсыру мерзімі: «24» мамыр 2021 ж.

Дипломдық жобаның бастапқы мәліметтері (зерттеу (жоба) нәтижелерінің талап етілген параметрлері мен объектінің бастапқы мәліметтері): Ұсынылып отырған дипломдық жобада Python программалау тілінің Django фреймворкін қолдана отырып, Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған веб-сайт әзірлеу. Жобаны орындау барысында Python, HTML, CSS тілдерін, ал мәліметтер қоры ретінде SQLite қолданамын.

Дипломдық жобада қарастырылған мәселелер тізімі немесе дипломдық жобаның қысқаша мазмұны:

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

Графикалық материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс):
1 диаграмма, 16 кесте, 25 сурет ұсынылған.

Ұсынылатын негізгі әдебиеттер:

1 Cameron Davidson-Pilon. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference // Addison-Wesley Data & Analytics, 2015.

2 UML – диаграмма классов // Сайттың электрондық нұсқасы
<https://prog-cpp.ru/uml-classes/>

3 Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.

4 Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

Дипломдық жобаның бөлімдеріне қатысты белгіленген кеңес берушілер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Боканова Г.Ш.		
Өміртіршілік қауіпсіздігі	Абрешов Ш.А.		
Программалық қамтама	Айтқазина Ә.А.		
Норма бақылау	Оралхан П.		

Дипломдық жобаны дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Талдау бөлімі		
Жобалау бөлімі		
Жобаны жүзеге асыру және тестілеу бөлімі		

Тапсырманың берілген күні «___» _____ 20__ ж.

Кафедра меңгерушісі _____ А.У. Утегенова

Жобаның ғылыми жетекшісі _____ О.Ж. Мамырбаев

Тапсырманы орындауға алған білім алушы _____ А.Қ. Мұхтария

Аңдатпа

Бұл дипломдық жоба «Әйелдер киімдеріне онлайн тапсырыс беру» тақырыбына арналған. Интернет-дүкен тауарлардың әртүрлі санаттарын ұсынады. Бұл дипломдық жобада MVC технология қолданылады. IDE интеграцияланған даму ортасы (Integrated Development Environment) PyCharm және SQLite деректер қорын басқару жүйесі осы жобада қолданылады. Әзірлеу кезінде Python программалау тілінің Django фреймворкы пайдаланылды. Бұл бізге статистикалық және динамикалық мазмұнды жасауға мүмкіндік береді. Сондай-ақ, біздің веб-қосымшаның логикасына жауап беретін модельдер қолданылды. Веб сайт екі бөліктен тұрады – серверлік және клиенттік.

Аннотация

Данный дипломный проект посвящен теме "онлайн заказ женской одежды". Интернет-магазин предлагает различные категории товаров. В данном дипломном проекте используется технология MVC. Интегрированная среда разработки IDE (Integrated Development Environment) PyCharm и система управления базами данных SQLite используются в этом проекте. При разработке использовался фреймворк Django языка программирования Python. Это позволяет нам создавать статистический и динамический контент. Также были использованы модели, отвечающие логике нашего веб-приложения. Сайт состоит из двух частей – серверной и клиентской.

Annotation

This diploma project is dedicated to the topic "online ordering of women's clothing". The online store offers various categories of products. This graduation project uses MVC technology. The Integrated Development Environment (IDE) PyCharm and the SQLite database management system are used in this project. During development, the Django framework of the Python programming language was used. This allows us to create statistical and dynamic content. We also used models that correspond to the logic of our web application. The site consists of two parts-server and client.

Мазмұны

Кіріспе	9
1 Талдау бөлімі	10
1.1 Қазіргі заманғы веб-қосымшаларды талдау және шолу	10
1.2 Веб-сервистердің түрлері	13
1.3 Қолданыстағы веб –қосымшаларды құру технологияларын талдау	14
2 Жобалау бөлімі	20
2.1 Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған веб-қосымшаның функционалдық моделі	20
2.2 Жүйенің жұмыс істеуінің жалпыланған алгоритмі	20
2.3 Әйелдер киіміне тапсырыс беру бойынша веб-қосымшаны ақпараттық қамтамасыз ету	22
2.4 Әйелдер киімдеріне онлайн тапсырыс беруді әзірлеу және іске асыру барысында қолданылатын технологиялар	27
2.5 UML модельдеу тілі	33
2.6 Тізбек диаграммасы	35
2.7 Прецеденттер диаграммасы	40
2.8 Класстар диаграммасы	42
3 Жүзеге асыру және тестілеу бөлімі	47
3.1 Әйелдер киімдеріне арналған веб-сайт	47
4 Экономикалық бөлім	51
4.1 Жобаның сипаттамасы	51
4.2 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу	51
4.3 Бағдарламалық қамсыздандыру шығынының есебі	53
4.4 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі	57
4.5 Игеру саласындағы жылдық бір жолғы шығындар есебі	57
4.6 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі	59
4.7 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу	60
4.8 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау	61
4.9 Экономикалық бөлім бойынша қорытынды	64
5 Өміртіршілік қауіпсіздігі	65
5.1 Жұмыс орнының эргономикасы және еңбек жағдайларын талдау	65
5.2 Жасанды жарықтандыруды нүктелік әдіспен есептеу	73
Қорытынды	79
Әдебиеттер тізімі	80
А Қосымшасы	81
Ә Қосымшасы	85

Кіріспе

21 ғасыр – ақпараттық технологиялар ғасыры. Бүкіл әлемде адамдар жаңа технологияларды ұстануда. Көбінесе адамдарға үйден шықпай-ақ кез-келген операцияны сатып алуға, сатып алуға, жасауға ыңғайлы және тиімдірек болады. Интернет адамдардың мүмкіндіктерін адамзат елестете алмайтын деңгейге дейін жетті. Интернет заманауи адамға дамуға (өзін-өзі дамыту), ойын-сауық, сондай-ақ бизнес саласында үлкен мүмкіндіктер береді. Егер бұрын адам қандай да бір сатып алу жасау үшін жол, кезектер және т.б. сияқты келесі әрекеттерге көп уақыт жұмсауға мәжбүр болса, онда ғаламдық желі ұсынатын қызметтер адамдарға іс-әрекет еркіндігін береді. Бизнес жағдайында бұл Google AdWords және Яндекс.Директ маркетингтік құралдарының заманауи әдістерін қолдана отырып ақша табудың тәсілі. Қазіргі жағдайда бизнестің табыстылығы мен сәттілігі көбінесе компанияның нарықтағы жағдайдың өзгеруіне жауап беретін жылдамдығына байланысты. Бұл фактор өнімділікпен бірге экономиканың әртүрлі салаларындағы кәсіпорындардың стратегиялық активіне айналады. Осыған байланысты алдыңғы қатарлы ақпараттық технологияларды қолдану сәттіліктің шешуші факторына, көшбасшылық үшін күрестегі басты көшбасшылыққа айналуға.

Осындай мүмкіндіктердің бірі – клиентке әртүрлі өнімдердің кең спектрін ұсынатын интернет-дүкен. Интернет-дүкен алғаш рет АҚШ-та 1994 жылы пайда болды. Интернет-дүкен ұғымын американдық кәсіпкер Джефф Безос енгізді, ол Amazon интернет-компаниясының негізін қалаушы, әртүрлі тауарлар мен қызметтерді сататын ең үлкен алаң. Кейінірек, 1997 жылы, АҚШ-тың Интернет-дүкеніне ұқсас, Ресейде интернет-дүкен пайда болды. Қазіргі уақытта Интернеттегі тауарларды сату бойынша әлемдегі ең үлкен интернет-сайттар – Amazon, AliExpress, Walmart, eBay, bpm, Carters сияқты алыптар. Қазір бүкіл әлемдегі адамдар интернет-дүкеннің қызметіне жүгінеді, өйткені бұл ыңғайлы және тиімді. Егер бұрын интернет-дүкен жарамдылық мерзімі жоқ тауарларды сату үшін пайдаланылған болса, қазір интернет-дүкендер арқылы сіз кез-келген нәрсеге тапсырыс бере аласыз. Клиенттермен тиімді қарым-қатынас орнату және бәсекеге қабілеттілікті арттыру көбінесе инновациялық технологияларды енгізуге байланысты.

Интернет-дүкеннің пайда болуымен көптеген адамдардың өмірі жеңілдеді, өйткені әлемнің түкпір-түкпірінен тауарларға тапсырыс беру мүмкіндігі пайда болды. Бұл қазіргі әлемнің жаңашыл бағдарламашыларының еңбегі. Қазіргі жағдайда интернет-дүкеннің тиімділігі мен сәттілігі нарықтағы жағдайдың өзгеруіне байланысты. Бірақ бұл тек тауарларды ұсынумен аяқталмайды, іс жүзінде интернет-дүкеннің идеясы жан-жақты: ол соңғы тұтынушы үшін ең қолайлы жағдай жасауды қарастырады.

1 Талдау бөлімі

1.1 Қазіргі заманғы веб-қосымшаларды талдау және шолу

Қазіргі заманғы веб-қосымшаның келесі сипаттамалары болуы керек.

Қауіпсіздік. Қазіргі әлемдегі веб-қосымшаның маңызды критерийлерінің бірі болып саналады. Пайдаланушылардың ешқайсысы веб-қосымшаның қызметтерін, егер оның жеке деректерінің қауіпсіздігіне кепілдік берілмесе, пайдаланбайды. Пайдаланушы өзінің құпия деректерін бөтен адамдар алады деп алаңдамауы керек. Веб-ресурстарға хакерлік шабуылдардың әртүрлі түрлері бар, біз ең көп кездесетіндерін тізімдейміз. SQL инъекциясы – бұл GET, POST сұраулары немесе cookie мәндері арқылы SQL серверіне дерекқордан кез-келген деректерді өзгерте, тарта немесе жоя алатын код берілетін хакерлік шабуылдың бір түрі. DoS шабуылы кез-келген бағдарламаны немесе жүйені істен шығару үшін қолданылады, содан кейін бағдарламаны немесе жүйені қайта іске қосу қажет болады. Вирустар-орташа пайдаланушы үшін ең танымал шабуылдар. Мәні енгізуге зиянды бағдарлама компьютер, ол содан кейін көмектеседі бекітілу барысында қаскүнем ұрлап қандай да бір құпия деректер. Вирустар орташа пайдаланушыға айтарлықтай зиян келтіруі мүмкін, бірақ антивирустың көмегімен біз жеке компьютерге әртүрлі шабуылдардың алдын аламыз.

Ауқымдылығы. Веб-қосымшаның маңызды өлшемі. Қазіргі әлемдегі серверлерге жүктеме өте үлкен, әр екінші адамда аптасына 7 күн, тәулігіне 24 сағат ғаламдық желіге кіруге мүмкіндік беретін әртүрлі құрылғылар (смартфон, ноутбук, планшет) бар. Пайдаланушылар веб-ресурсқа риза болуы үшін бағдарламалық жасақтаманың да, аппараттық құралдың да ресурстарын көбейтуге мүмкіндік беретін масштабталу сияқты сипаттамаға назар аудару керек. Масштабтаудың екі түрі бар: көлденең және тік. Көлденең масштабтау қосымшаның жұмысын жақсартатын құйрықтардың санын көбейтуді білдіреді. Тік масштабтау жад, процессор, жедел жад және т. б. сияқты ресурстарды қосу арқылы веб-қосымшаның жұмысын арттырады.

Икемділік. Бастапқы кодтың кез-келген бөлігін толықтырған немесе өзгерткен кезде қосымшаның құрылымы өзгермеуі үшін веб-бағдарлама икемді болуы керек. Сондай-ақ, әзірлеуші әрқашан осы технологияларды өз жобасына енгізе алатындай веб-технологияларды дамыту сияқты факторды ескеруіміз керек. Әзірлеу әрқашан топтық жұмысты қамтиды, онда жобаның икемділігі веб-әзірлеушінің қолданылатын құралдарына байланысты болады.

Ақауларға төзімділік және қол жетімділіктің жоғары деңгейі. Сондай-ақ, маңызды критерий-бұл ақауларға төзімділік және веб-қосымшаның қол жетімділігінің жоғары деңгейі. Веб-ресурс қателер мен уақытқа қарамастан пайдаланушылар үшін әрқашан қол жетімді болуы керек.

Кез-келген веб-бағдарлама-бұл тұрақты және динамикалық веб-беттер жиынтығы. Тұрақты веб-бет-бұл әрқашан пайдаланушының алдында өзгермейтін түрде көрсетілетін бет. Веб-сервер бетті веб-шолғыштың

сұранысы бойынша ешқандай өзгеріссіз жібереді. Керісінше, сервер браузерге жібермес бұрын динамикалық веб-параққа өзгертулер енгізеді. Бет өзгеріп жатқандықтан, ол динамикалық деп аталады. Мысалы, сіз сауықтыру бағдарламасының нәтижелері көрсетілетін бетті жасай аласыз. Бұл жағдайда кейбір ақпарат (мысалы, қызметкердің аты және оның нәтижелері) қызметкер бетті сұраған кезде анықталады. Тұрақты веб-сайтта веб-сервер орнатылған компьютерде орналастырылған тиісті HTML парақтары мен файлдар жиынтығы бар.

Біздің елімізде дәстүрлі бөлшек сауда орындары тауарларға еркін қол жетімділік қағидатына негізделген дүкендермен ауыстырылған кезде, бұл тұтынушылық белсенділіктің күрт өсуіне әкелді.

Енді тұтынушы өздерін қызықтыратын өнімді әлемнің кез-келген нүктесінен екі рет басу арқылы таба алады. «Тұтынушы жағдайдың қожайынына айналсын» деген ең маңызды ережені қабылдап, сатып алу процесін мейлінше ыңғайлы етуге тырысып, мұндай дүкендердің иелері сатылымның айтарлықтай өсуіне қол жеткізді.

Интернет-дүкендер, шын мәнінде, осы қағидаларды жүзеге асырудың келесі қадамы болды. Оларда сатушы өзінің ұсақ-түйектері мен қылығына мойынсұнып, тұтынушыны әрдайым қуантуға тырысады. Мұндай жағдайда жұмыс істеу қиынырақ. Алайда, қиындықтар интернет-дүкендердің сатыларды тапсырыс беруден бастап жеткізуге дейін біртіндеп, кезең-кезеңімен автоматтандыру мүмкіндігімен өтеледі.

Нәтиже көп күттірмейді - сатып алушы сатып алудың төмен бағалары мен ыңғайлылығын бағалайды. Сонымен бірге, сауданы толық автоматтандыру бизнесті одан әрі сәтті дамыту үшін берік негіз қалайды, бұл ескі көне бәсекелестерден озуға мүмкіндік береді. Сатып алушының дәстүрлі дүкен қызметтерінен бас тартуына және интернет-дүкенді таңдауға себеп болатын факторлар: дәстүрлі дүкендермен салыстырғанда төменірек бағалар (интернет-дүкенде бөлшек сауда орындарын жалдауға және көп қызметкерлер жалдауға міндетті болып келмейді. Бұл бағаны біршама төмендетуге мүмкіндік береді. Клиент ешқашан қосымша ақы төлегісі келмейді); уақытты үнемдеу (сатып алушыға ешқандай барудың қажеті жоқ, ол күннің кез келген уақытында сатып ала алады. Бұл, сөзсіз, сатушы үшін пайдалы, өйткені ол мүмкін тапсырыстар санын көбейтеді. Клиентке тауарды көрсетілген жерге жеткізу ыңғайлы және уақытты үнемдейді); керекті нәрсені ыңғайлы, жылдам және тиімді іздеу (зерттеулер көрсеткендей, интернет-дүкендердің клиенттері әдетте нақты бір нәрсені іздейді. Егер тауарлардың бар-жоғы үнемі тексеріліп отырса, бағалар жаңартылса және сайтта іздеу функциясы болса, онда сатып алушыны алуға барлық мүмкіндік бар); осы дүкеннің басқа тұтынушыларымен пікір алмасу мүмкіндігі (тұтынушы өздері іздегенге ұқсас нәрсені іздейтін адамдармен қарым-қатынас жасауды қалайды. Егер ол затты қазір және қазір сатып алмаса да, пайдалы кеңестер форумда оны қайта-қайта келуге мәжбүр етеді және мұнда бәрі дүкен мен қызметкерлерге байланысты); Егер жоғарыда аталған барлық шарттар орындалса, көп ұзамай офлайн дүкендер аз қалады. Кең және

үнемі өсіп келе жатқан аудитория (географиялық шектеудің болмауы өте маңызды - желі барлығына ашық, жеткізілім шарттары шектеулі. Алайда, көбінесе, егер сатушы көтерме саудамен айналысса немесе сирек кездесетін тауарларды сатса, сатып алудың пайдасы тасымалдау құнынан едәуір асады); бөлшек сауда орындарын, құрал-жабдықтарды жалдау және қызметкерлерді жалдау құнын төмендету (интернет-дүкен, әдетте, қызметкерлердің аз санына қызмет көрсете алады); кәсіпорынды автоматтандыру (автоматтандыру өңделген тапсырыстар санын көбейтуге және қателіктер ықтималдығын азайтуға мүмкіндік береді, жұмысшылардың санын тағы да қысқартады және барлық ақпаратты бір жүйеге біріктіру арқылы кәсіпорынды тиімді басқаруға мүмкіндік береді. Нәтижесінде - офлайн бәсекелестерден айтарлықтай артықшылық); тұтынушылар туралы маркетингтік ақпаратты жинаудың ыңғайлылығы (алынған статистикалық мәліметтер: түрлі сауалнамалар, сауалнамалар, есептегіш көрсеткіштері - нарық қажеттіліктеріне тез жауап беруге мүмкіндік береді); жарнамалық шығындар азаяды (бұл дүкендердің белгілі бір санаттары үшін үлкен артықшылық. Барлық тауашалар офлайн режимінде болған және сапалы науқан жүздеген мың немесе миллиондаған долларды қажет етсе, интернетте, салыстырмалы түрде жарнама бюджеті болады, және әсерлі нәтижелерге алаулы қол жеткізуге болады).

Интернет-дүкен - сатып алушыға интернет арқылы сатып алу үшін қажетті ақпаратты, оның ішінде өнімдердің түрлерін, бағаларын, сатушысын, төлем мен жеткізудің әдістері мен шарттарын, сонымен қатар сатып алушыға ақпарат беруге арналған сауда ұйымының бөлігі. Интернет арқылы сатып алушылардан тауарларды сатып алу ниеті туралы хабарламалар алуға, сондай-ақ сатушы немесе оның мердігері сатып алушы көрсеткен мекен-жайға немесе қабылдау пунктіне жеткізілуіне кепілдік береді.

Интернет-дүкенді ұстау құны дәстүрлі бөлшек сауда дүкенін ұстауға жұмсалған шығындардан едәуір төмен. Біз үй-жайларды сатып алуға немесе жалға беруге және еңбек шығындарын үнемдеуге қатысты боламыз. Сату арнасы ретінде интернеттің артықшылықтары Қазақстанның ірі сауда желілерінде де айқын болды. Өткен жылы тұрмыстық техника мен электрониканың бөлшек сауда желілері интернет-бизнесті кеңейтуде ерекше белсенді болды.

Интернет-дүкеннің логикасы нақты сауда алаңының қағидаттарына негізделген. Басқаша айтқанда, кез-келген интернет-дүкен құрылымдық өнімдер каталогын (дүкендегі сөрелерге ұқсас), кілт сөздерді іздеу жүйесін (сату аймағындағы бөлімдерге ұқсас), тапсырыс себеті (тұтынушы таңдаған нақты себетке ұқсас), төлем жүйелерінің әртүрлілігі және жеткізу тәсілдері мен қызметтерін таңдау.

Анықтамада 4 негізгі ой бар:

- интернет-дүкен - бұл веб-сайттың өзі емес, бизнес немесе оның бөлігі;
- сатып алушы сатып алу үшін қажетті ақпаратты веб-сайт арқылы алады;
- сатып алушы интернет арқылы сатушыға хабарлауы мүмкін;

- интернет-дүкен (өзі немесе мердігер арқылы) кем дегенде қабылдау пунктіне жеткізуді қамтамасыз етеді.

Интернет-коммерция - бұл сауда-саттықтың жеке және тәуелсіз аймағы емес, ол тек оның формаларының бірі. Сатып алушы өз таңдауын оған неғұрлым қолайлы жағдайлар жасайтын сатушының пайдасына жасайды. Ешкім дүкенге тек желіде болғандықтан бара алмайды, сондықтан интернет-дүкендер желі беретін артықшылықтарды түсінуге тырысуы керек.

Қазақстанда сауда жүргізетін интернет-дүкендер бар. Олар: technodom.kz, alser.kz. Бұл сайттарға қысқаша анықтама беретін болсақ, бұл сайттар электроника саласындағы интернет-дүкендер. Бұл сайттарда кез-келген электроника заттарымен танысып, сатып алуға болады. Дегенмен қазіргі уақытта көлік немесе пәтер сияқты ерекшелену керек заттарда бар. Олар әрине электроникаға қатысты тауарлар. Анығырақ айтсақ, байланыс телефондар және компьютерлер. Статистика бойынша кез-келген екінші адамда ұялы телефон немесе компьютер бар. Өйткені заманның талабы солай. Тіпті жоғары айтып өткен technodom.kz, alser.kz секілді сайттардың өзіне кіру үшін бізге ғаламторға кіру мүмкіндігі бар ұялы телефон, компьютер немесе ақылды теледидарлар қажет болады. Бұдан бөлек адамдар өзара байланысу үшін, бір-бірімен хабар алмасу немесе жаңалықтармен бөлісу үшін де ұялы телефондарды сатып алады.

Қазіргі таңда біздің өмір сүріп жатқан заманымыз технология заманы болғандықтан мен электроника сататын сауда компаниясының интернет-дүкенін құруға негізделдім. Менің жасалынатын сайтымда, тек қана электроника және оларға арналған қызметтер болады.

1.2 Веб-сервистердің түрлері

Веб-қызметтердің әртүрлі түрлері бар. Олардың әрқайсысы өзінің нақты міндетін шешеді.

Корпоративтік портал. Корпоративтік порталдар бизнес-процесті оңтайландыру үшін ұйымдарды жиі пайдаланады. Ұйым қызметкерлері порталды қосымшаларға немесе қажетті деректерге қол жеткізу үшін пайдалана алады. Корпоративтік порталды өзара іс-қимыл жасай алатын барлық қызметкерлер үшін бірыңғай ақпараттық алаң ретінде интернетпен ассоциациялауға болады. Корпоративтік порталдардың артықшылықтары өте көп, біз кейбіреулерден өтеміз. Ұйым қызметкерлері өзара қарым-қатынас жасау үшін әртүрлі форумдар құра алады, әртүрлі жаңалықтар мен материалдармен бөліседі, сонымен қатар маңызды тапсырмаларды орындау кезінде қолданылатын әртүрлі файлдар мен құжаттарды жасай алады. Қызметкерлердің ұтқырлығы мен жұмыс қабілеттілігін арттыру.

CRM (Customer Relationship Management). Жүйе тұтынушылармен өзара әрекеттесуді жеңілдетеді. Клиенттерді басқару жүйесі тұтынушылар туралы қажетті ақпаратты, Сату көрсеткіштерін жақсарту және бизнес-процесті жеделдету үшін олармен өзара әрекеттесу тарихын жинайды. Клиенттер,

серіктестер, жеткізушілер туралы ақпарат жинау және деректерді кейіннен талдау бизнеске ең тиімді режимде жұмыс істеуге көмектеседі.

CRM жүйелерінің негізгі принциптерін бөліп көрсетуге болады: бизнес-процестерді оңтайландыру, деректерді бірыңғай сақтау, қоңыраулар, конгрестер, әлеуметтік желілер, пошта және т.б. сияқты арналарды пайдалану.

ERP жүйелері. Кәсіпорын ресурстарын жоспарлау. Бұл бағдарламалық жасақтама кәсіпорын ресурстарын барынша тиімді пайдалануға көмектеседі. ERP жүйесі кәсіпорындағы процестерді үндестіруге және басқаруға көмектеседі, сондай-ақ компания бөлімдері арасындағы өзара іс-қимылды жақсартады және еңбек ресурстарын үнемдейді.

ITRP (IT Resources Planning) — корпоративтік АТ-ресурстары мен сервистерді басқаруды қолдауға арналған Web-қосымшалар класы.

OSS (Operation Support Systems) — web-қосымшалардың бір түрі, таратылған есептеу желілерінің операторларының жұмысын қамтамасыз етуге арналған. OSS желіні басқаруды, өнімділікті, сәтсіздіктерді жоюды, қызметтерді құру мен есепке алуды, желілік ресурстарды жоспарлауды, процестерді бақылауды, қауіпсіздікті бақылауды, қызмет сапасы мен клиенттерге қызмет көрсету деңгейін, статистикалық мәліметтерді жинауды қамтамасыз етеді. Oss — тің бір түрі-бизнесі қолдау жүйесі-BSS (Business Support Systems). Оларға есеп айырысу жүйелері, клиенттермен қарым-қатынасты басқару жүйелері, желілерді басқару, тапсырыс беру, қызмет көрсету сапасы кіреді.

Мамандандырылған веб-сайттар. Бұл сайттардың өз тақырыптары мен мақсатты аудиториялары бар. Мамандандырылған веб-сайттарға: блог, интернет-дүкен, әлеуметтік желі, форум, ойын сайттары және т.б. әлеуметтік желілер интернеттегі адамдармен сөйлесу үшін қолданылады. Сондай-ақ, әр пайдаланушы өзі туралы Жеке ақпараты бар профиль жасай алады. Түрлі фотосуреттер, бейнелер мен жаналықтарды таратыңыз. Әлеуметтік медиа бизнес үшін де, ойын-сауық үшін де қолданылады.

Интернет-дүкендер көптеген адамдардың өмірін жеңілдетті. Егер бұрын адам кез-келген әрекетті орындау үшін көп уақыт пен күш жұмсауға тура келсе, онда Интернет-дүкендердің пайда болуымен адамның мүмкіндіктері кеңейе түсті. Клиенттерге әлемнің кез-келген жерінде практикалық түрде алуға болатын көптеген өнімдер ұсынылады.

1.3 Қолданыстағы веб –қосымшаларды құру технологияларын талдау

Веб-қосымшамды жасау үшін мен жоғары деңгейлі Python бағдарламалау тілін қолдандым. Python тілі-императивті, процедуралық, құрылымдық, объектіге бағытталған бағдарламалауды, метапрограммалауды және функционалды бағдарламалауды қолдайтын көп парадигмалық бағдарламалау тілі. Даму ортасы ретінде мен Python тілінің даму ортасы болып табылатын Pycharm-ді қолдандым. Бұл орта веб-қосымшаның кодын мүмкіндігінше

ыңғайлы жазуға мүмкіндік береді. Сондай-ақ, ол даму процесін мүмкіндігінше жеңілдетуге мүмкіндік беретін жылдам пернелер деп аталады.

Кейбір объектілі-бағытталған бағдарламалау тілдері объектілердің сериализациясын (немесе объектілерді мұрағаттауды) немесе синтаксистік элементтердің көмегімен немесе бұл үшін стандартты интерфейсті қамтамасыз ету үшін тікелей қолдайды. Осы бағдарламалау тілдерінің кейбіреулері Ruby, Smalltalk, Python, PHP, Objective-C, Delphi, Java және тілдер отбасысы болып табылатын. NET. Сондай-ақ, кірістірілген қолдау жоқ тілдер үшін сериализацияға қолдау көрсететін кітапханалар да бар. CFML деректер құрылымын тегпен және JSON-ның SerializeJSON() функциясы арқылы WDDX деректер құрылымын сериализациялауға мүмкіндік береді. OCaml стандартты кітапханасы Marshal модулі (оның құжаттары) және Pervasive output_value және input_value функциялары арқылы маршалингті қамтамасыз етеді. OCaml бағдарламалау тілі типке статикалық түрде тексерілген кезде, Marshal модулі пайдалану тип кепілдігін бұзуы мүмкін, себебі тексеру тәсілі жоқ болғандықтан, күтілген түрдегі объектілер маршалингтелмеген ағын ұсынады ма? OCaml функцияны немесе функцияны 13 қамтитын деректер құрылымын (мысалы, әдіс бар нысан) бағыттау қиын, себебі функциялардағы орындалатын код әр түрлі бағдарламалар арқылы берілуі мүмкін емес. (Маршалинг коды позициясы функциясының туы бар, бірақ ол тек сол бір бағдарламада ғана жойылуы мүмкін). Стандартты маршалинг функциялары ортақ қатынауды сақтап, түмен теңшеуге болатын циклдік деректерді өңдей алады. Кейбір Perl модульдер CPAN механизмдердің сериализациясын қамтамасыз етеді, оның ішінде Storable, JSON: XS және FreezeThaw. Storable Perl деректер құрылымын файлдар мен файлдар немесе Perl скалярлары үшін сериализация және десериализациялау функцияларын қамтиды. Файлдарды тікелей сериализациялаудан басқа, Storable скаляр оралған деректердің іске асырылмаған көшірмесін қайтару freeze функциясын және мұндай скалярдың thaw десериализациясын қамтиды. Бұл желілік сокет арқылы күрделі деректер құрылымын жіберу немесе оны деректер базасында сақтау үшін пайдалы. Storable арқылы құрылымдарды сериализациялау кезінде желілік қауіпсіз функциялар бар, олар әрқашан өз деректерін аз жылдамдықпен кез келген компьютерде оқылатын пішімде сақтайды. Бұл мүмкіндіктер N-store, N-freeze және т. б. аталды. Бұл құрылымдарды десериализациялау үшін "n" функциялары жоқ - "N" функцияларымен және олардың машинаға тәуелді баламаларымен сериализацияланған, десериализацияланған тұрақты thaw және retrieve құрылымдар. C және C++ қандай да бір жоғары деңгейлі құрылым ретінде сериализацияны қамтамасыз етпейді, бірақ екі тіл де кірістірілген деректердің кез келген түрін, сондай-ақ екілік деректер түрінде қарапайым ескі деректер құрылымын жазуды қолдайды. Осылайша, сериализацияның таңдамалы функцияларын жазу әдетте тривиальды. Сонымен қатар, C++ үшін ODM, ORM жүйесі сияқты шешімдер компиляторында негізделген, шағын өзгерістермен немесе сынып хабарландыруларында өзгеріссіз сериализация кодын автоматты түрде жасауға қабілетті. Басқа танымал сериализация

фреймворктары - Boost.Boost Framework бастап Serialization, s11n фреймворк, және Cereal. Microsoft көп функциялы орталықтың құрылымы да өзінің Document-View архитектурасының бір бөлігі ретінде сериализация әдіснамасын ұсынады.

Python - жалпы мақсаттағы бағдарламалау тілі. Python ядросының синтаксисі ең аз. Сонымен қатар стандартты кітапхана пайдалы функциялардың үлкен көлемін қамтиды. Python құрылымдық, объектілі-бағытталған, функционалдық, императивті және аспектілі-бағытталған бағдарламалауды қолдайды. Негізгі архитектуралық ерекшеліктер - динамикалық типизация, жадыны автоматты басқару, толық интроспекция, ерекшеліктерді өңдеу механизмі, көп ағынды есептеулерді қолдау, деректердің жоғары деңгейлі құрылымы. Өз кезегінде, пакеттерге бірігуі мүмкін модульдерге бағдарламаларды бөлу қолдайды. Python динамикалық типтеуді қолдайды, яғни айнымалы түрі тек орындау кезінде анықталады. Сондықтан "айнымалы мәнін тағайындау "орнына" кейбір атауымен мәнді байланыстыру " туралы айту жақсы. Pythonда енгізілген түрлері бар: түйреуіш, жол, Unicode-жол, еркін дәлдіктің бүтін саны, өзгермелі үтірмен Сан, кешенді сан және т.б. Python коллекциясынан енгізілген: тізім, кортеж (өзгермейтін тізім), сөздік, көптеген және басқалар. Барлық мәндер объектілер болып табылады, оның ішінде функциялар, әдістер, Модульдер, сыныптар.18 Жаңа түрді қосу класты (class) жазу немесе кеңейту модулінде жаңа түрді (мысалы, C тілінде жазылған) анықтау арқылы болады. Кластар жүйесі мұраға (жалғыз және көпше) және метапрограммалауға қолдау көрсетеді. Көптеген кірістірілген түрлер мен кеңейтулердің типтерінен мұрагерлік болуы мүмкін. Барлық объектілер сілтемелік және атомдық болып бөлінеді. Атомдық int, long (3 нұсқасында int кез келген саны, себебі 3 нұсқасында өлшемге шектеу жоқ), complex және басқа да. Атомдық объектілерді беру кезінде олардың мәні көшіріледі, ал сілтемелік үшін тек объектіге көрсеткіш көшіріледі,осылайша, берілгеннен кейін екі айнымалылар бірдей мәнді пайдаланады. Сілтеме нысандары өзгертілетін және өзгермейтін болады. Мысалы, жолдар мен кортеждер өзгермейтін, ал тізімдер, сөздіктер және басқа да көптеген нысандар өзгертін болады. Python-да Кортеж шын мәнінде өзгермейтін тізім. Көптеген жағдайларда кортеждер тізімдерден жылдамырақ жұмыс істейді, сондықтан сіз тізбекті Өзгертуді жоспарламасаңыз, онда оларды пайдалану жақсы. Python-белсенді дамып келе жатқан бағдарламалау тілі, тілдік қасиеттерді қосу/өзгерту жаңа нұсқалары шамамен екі жарым жылда бір рет шығады. Тіл ресми стандарттауға ұшырамады, де-факто стандартының рөлін Тіл авторының бақылауымен жасалған CPython орындайды [10]. Python қарапайым екілік кодтау декодеріне негізделген (SBE). Sbedecoder - SBE кодталған деректерді талдау үшін қарапайым python пакеті. sbedecoder XML пішіміндегі SBE парсерін динамикалық жасайды. Бұл sbeschema() данасын жасау және файл атымен оның parse() әдісін шақыру арқылы қол жеткізіледі: from sbedecoder import SBESchema schema = SBESchema() schema.parse('path/to/schema.xml'). Бұл объектілер мен деректердің сериализациясын тікелей қолдайтын объектілі-

бағытталған бағдарламалау тілдерінің бөлігі болды. Python әзірлеу үшін бірнеше арнайы IDE бар. JetBrains PyCharm-Python үшін толық функциялы IDE, Windows, Mac OS X және Linux платформаларында қол жетімді, тегін (Community) және ақылы (Professional) нұсқаларда бар. PyCharm-Python бағдарламалау тілі үшін интеграцияланған даму ортасы. Кодты талдау құралы, графикалық реттеуші, юнит-тестерді іске қосу құралы және Django веб-әзірлемесін қолдайды. PyCharm IntelliJ IDEA негізінде JetBrains компаниясымен жасалған.

PyCharm Windows, Mac OS X және Linux операциялық жүйелерімен жұмыс істейді.

Мүмкіндіктері:

- Кодты статикалық талдау, синтаксис және қателерді жарықтандыру.
- Жоба және бастапқы код бойынша Навигация: жобаның файлдық құрылымын көрсету, файлдар, сыныптар, әдістер мен әдістерді пайдалану арасында жылдам өту.
- Рефакторинг: атауын өзгерту, әдісті шығару, айнымалыны енгізу, константаны енгізу, әдісті көтеру және түсіру және т. б.
- Django фреймворк пайдалану веб-әзірлеу құралдары
- Python үшін кірістірілген реттеуші
- Юнит-тестілеу үшін кірістірілген құралдар
- Google App Engine пайдаланып әзірлеу
- Нұсқаларды бақылау жүйесін қолдау: Mercurial, Git, Subversion, Perforce және CVS үшін жалпы пайдаланушы интерфейсі.

Даму барысында бағдарламашы бірнеше қателермен бірнеше рет кездеседі, бірақ PyCharm idea бұл қателерді анықтауға және оларды бастапқы кезеңде салдарсыз жоюға көмектеседі. Ол үшін debug деп аталатын арнайы түзету бағдарламасы бар, ол сізге бағдарламаның дұрыстығын тексергіңіз келетін жерде кодты тоқтатуға мүмкіндік береді. Егер бұрын қосымшаны әзірлеу кезінде бағдарламашы бастапқы кодты үнемі сақтап тұруға мәжбүр болса, содан кейін командалық жол арқылы бастапқы кодты компиляторға қоңырау шалса, онда PyCharm әзірлеу ортасы қажетсіз әрекеттерді жояды. Бұл орта сіздің жобаңызды бастапқы кодты толықтырған немесе өзгерткен кезде одан әрі ыңғайлы болу үшін құрылымдауға мүмкіндік береді. Сондай-ақ, қоршаған орта GIT сияқты қуатты даму құралын пайдалануға мүмкіндік береді, бұл құралсыз сіз жобаны басқа әзірлеушілермен бірлесіп жүзеге асыра алмайсыз. Git бағдарламасы Сіздің жобаңыздың құрылымын өзгертпестен бастапқы кодты өзгертуге және толықтыруға мүмкіндік беретін жаңа филиал құруға мүмкіндік береді. Сіз әрқашан әзірлеуші қандай қосымша енгізгенін немесе бастапқы кодты қалай өзгерткенін көре аласыз. Сіздің жобаңызда жазу үшін бірнеше жобаларды біріктіру және ең жақсы бастапқы кодты анықтау мүмкіндігі бар. Сондай-ақ, кез-келген жобаны пәрмен жолы мен git clone пәрмені арқылы клондауға болады.

Бағдарламалау тілі таңдалғаннан кейін Python веб-қосымшаларды құру үшін жақтауды таңдау керек.

Негізінен, әзірленген қосымшаны пайдаланушылар браузер арқылы жасалған қосымшамен жұмыс істей алуы үшін негіз қажет. Жақтау бағдарламалық жасақтаманы әзірлеуді жеңілдетеді және бағдарламалық жасақтама жүйесінің әртүрлі компоненттерін біріктіреді.

Жақтауды таңдау үшін Python тілінде веб-қосымшаларды әзірлеудің қолданыстағы құралдарын талдау қажет. Бірнеше шешім бар: Django, Flask, Pyramid және web2py. Алайда, Django және Flask әзірлеушілер арасында ең танымал, сондықтан дәл осы шеңберлерді талдау және салыстыру қажет.

Flask 2010 жылы австриялық әзірлеуші Армин Ронахер жасаған. Django бұл Django Software Foundation әзірлеген және 2005 жылы шығарылған жақтау. Айта кету керек, екі жақтау да ашық бағдарламалық кодпен, ақысыз.

Flash пен Django арасындағы негізгі айырмашылықтарды атап өту керек. Біріншіден, Django көбінесе күрделі веб-қосымшаларды құруда қолданылады, ал Flask бір немесе екі ерекшелігі бар шағын веб-қосымшаларға арналған.

Екіншіден, жаңа даму құралдарымен танысу кезінде олардың құжаттамасына назар аудару керек. Django құжаттамасы Flask құжаттамасынан әлдеқайда көп және көп. Бұл Django үйрену процесі түсінікті және жеңіл болады дегенді білдіреді.

Үшіншіден, Django – да әр түрлі мәліметтер базасымен жұмыс істеуге мүмкіндік беретін ORM-виртуалды Объектілік мәліметтер базасы бар. Flask-тің sqlalchemy сияқты үшінші тарап ORM-ді қосу мүмкіндігі бар, бірақ дайын шешімді қолдану әрқашан ыңғайлы.

Төртіншіден, Django көмегімен қосымшаны құру кезінде қатаң құрылымдалған каталог жүйесі автоматты түрде жасалады. Flask көмегімен әзірлеуші өзінің қосымшасының құрылымын жасайды. Бұл жағдайда Flask пайдалану ыңғайлы және оңай болып көрінуі мүмкін, бірақ олай емес. Flask көмегімен көптеген функциялары бар күрделі қосымшаны құру жағдайында, осы мақсат үшін арнайы құралдар болмаған кезде кодты құру әлдеқайда қиын.

Django-ның ең күшті артықшылықтарының бірі – автоматты түрде жасалған дерекқор әкімшісінің интерфейсі. Осы интерфейссті қолдана отырып, сенімді пайдаланушылар веб-қосымшаның дерекқорын ыңғайлы басқара алады.

Django және Flask салыстырудан Django жақтауы мәселені шешуге әлдеқайда жақсы екенін көруге болады, сондықтан ол веб-қосымшаны жасау кезінде қолданылады.

Әзірлеу құралдарын таңдаудағы келесі қадам-дерекқорды басқару жүйесін таңдау. Әдепкі бойынша, Django жақтауы SQLite ДҚБЖ қолданады. Бағдарлама SQLite қолданған кезде, интеграция интерфейс арқылы деректерді бөлісудің орнына, деректерді қамтитын файлға жасалған Функционалды және тікелей қоңыраулармен жұмыс істейді. Нәтижесінде, басқа ДҚБЖ-мен салыстырғанда серверге жүктеме жеңілдейді. Бұл SQLite-ді кітапхананың

негізгі технологиясының арқасында өте жылдам және тиімді етеді. Сонымен қатар, SQLite ең танымал ДҚБЖ ондығына кіреді.

2 Жобалау бөлімі

2.1 Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған веб-қосымшаның функционалдық моделі

Әйелдер киіміне онлайн тапсырыс беру үшін веб-қосымшасының функционалдық моделі екі модульдің – әкімші мен пайдаланушының өзара іс-қимылын білдіреді, қосымша мынадай функцияларды қамтиды:

Есептік жазбаларды жасау және өңдеу. Функция клиенттердің есептік жазбаларын қосуға және жоюға мүмкіндік береді. Деректерді енгізу үшін пайдаланушының электрондық пошта мекенжайын енгізу керек, ол кейіннен тіркелгі логині ретінде пайдаланылады және пайдаланушы бірінші авторизация кезінде өзгертуі керек уақытша құпия сөз.

Клиентке әрі қарай көрсету үшін дерекқорға өнімдерді қосу. Тауарлардың бағасы мен санының өзгеруі. Тауарлардың жаңа санаттарын толықтыру, қажет емес тауарларды алып тастау және т. б.

2.2 Жүйенің жұмыс істеуінің жалпыланған алгоритмі

Веб-қосымшаны әзірлеудің мақсаты – әйелдер киімін сатуға арналған веб-сайт құру, кез-келген киімге тапсырыс беру мүмкіндігі.

Даму екі модульден тұрады:

- "әкімші" модулі»;
- "пайдаланушы" модулі.

"Әкімші" модуліне басқа ішкі Модульдер кіреді:

- пайдаланушының аутентификациясы-дерекқорда пайдаланушының болуын тексеру;
- пайдаланушыны авторизациялау-пайдаланушының кіру құқығын тексеру және оның рөлдерін анықтау;
- пайдаланушыны қосу-әртүрлі рөлдермен пайдаланушыларды тіркеу (әкімші немесе пайдаланушы);
- қосалқы модуль тауарларды қосу-жаңа өнімдерді қосуға немесе бар тауарлардың сипаттамасын өзгертуге мүмкіндік береді;
- ішкі модуль Санат қосу-жаңа Санаттар қосу немесе бұрыннан барларын өзгерту үшін ішкі модуль;
- ішкі модуль тапсырысты өңдеу-клиенттің белгілі бір тапсырысын өңдеуге арналған ішкі модуль.

"Клиент" модулі келесі Ішкі модульдерді қамтиды:

- клиенттің аутентификациясы-дерекқорда пайдаланушының болуын тексеру;
- подмодуль себеті-пайдаланушы себетіне өнімдерді қосу үшін;
- ішкі модуль тауарларды қарау-сайтта берілген өнімдердің тізімін көруге мүмкіндік береді;

- құралдар тақтасының ішкі модулі - пайдаланушының сайтты оңай шарлау үшін;

- тауарларға тапсырыс-пайдаланушы ассортименттегі кез-келген тауарға тапсырыс бере алады;

- қосалқы модуль бөлшектері-белгілі бір өнім туралы барлық қажетті ақпаратты білуге мүмкіндік береді.

Бұл қосымшаны әзірлеу үшін MVC архитектурасы (Model-View-Controller, «Model-view контроллері») қолданылады. MVC қосымшаның деректерін, қолданушы интерфейсін және басқару логикасын үш тәуелсіз компонентке бөледі: модель, көрініс және контроллер.

Модель-мәліметтерді береді және бизнес логиканы ұсынады. Модельдік объектілер мәліметтер базасының моделінің күйін алады, сақтайды және күйлерін өзгерту арқылы контроллер командаларына жауап береді.

Көрініс - бұл пайдаланушы интерфейсі. Модельдердің өзгеруіне жауап ретінде пайдаланушыға модель деректерін көрсетуге жауап береді.

Контроллер пайдаланушының сұраныстарын өңдейді. Әдетте, пайдаланушы интерфейспен өзара әрекеттеседі және өз кезегінде URL сұранысы пайда болады. Сұрауды контроллер өңдейді. Контроллер модель деректерін көрсетеді және көрініс ретінде жауап көрсетеді.

MVC үлкен кодты бөліктерге бөлу арқылы жеңілдетуге мүмкіндік береді. Ұзын кодты жазу шатастырады және қатесіз өзгерту қиын. MVC нақты бағдарламалау тіліне тәуелді емес және объектіге бағытталған әдісті немесе басқа мысалдарды қажет етпейді. Мұнда бөлу үш файл (немесе файлдары бар үш қалта), модель, көрініс және контроллер қажет екенін білдірмейді. Іс жүзінде, модель қосымшаның негізгі көлемін алады және әр түрлі кластар, қызметтер, ДҚ сыныптары түрінде ұсынылады, сонымен қатар сабақтың әр түрі үшін жеке папкалар жасалады.

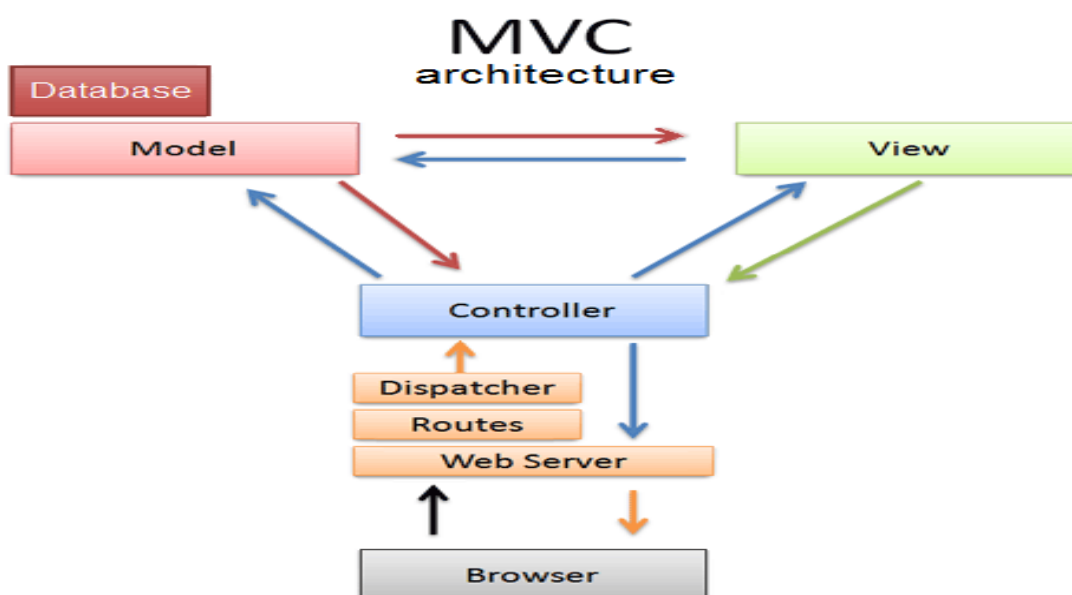
MVC көптеген қосымшаларға, соның ішінде серверлік веб-қосымшаларға және desktop (клиент) қосымшаларына қолданыла алады. Олардың арасындағы айырмашылық - веб-қосымшаның бағдарламасы пайдаланушының сұранысына жауап ретінде өңдейді, нәтижені көрсетеді (әдетте веб-парақ) және жабылады. Басқа сұраулар түскен кезде, бағдарламаның жаңа тәуелсіз көшірмесі өңделеді. Веб-қосымшалардың, дербес компьютерлерден және мобильді қосымшалардан (және JavaScript-те жасалған браузер беттерін қолданатын бағдарламалардан) айырмашылығы, олар ұзақ қызмет етеді. Олар пайдаланушының әртүрлі сұрауларын өңдейді және жүйеден шықпай-ақ ақпаратты жаңартады.

MVC компоненттерінің өзара әрекеттестігі серверде және desktop қосымшаларында біршама басқаша жүзеге асырылады, веб-қосымша бір пайдаланушының сұранысын өңдейді және аяқталады, ал desktop қосымшасы көптеген сұраныстарды қайта қоспай өңдейді.

Серверлік қосымшалар әдетте «пассивті» модельді қолданады, ал desktop бағдарламаларында «белсенді» модель қолданылады. Белсенді модельдерден айырмашылығы, пассивті модельдер өзгерістер туралы жазылуға және

хабарландырулар алуға мүмкіндік береді. Серверлік қосымшалар бұны қажет етпейді.

Серверлік қосымшалар пассивті модель жүйелерін қолданады. Мысалы, пайдаланушы форум парағына кіреді делік. Шолғышта ол HTTP сұранысын жібереді, ол хабарламалар тізімі бар парақты іздейді. Осы уақытта контроллер іске қосылады, пайдаланушының сұранысын талдайды және Model-дан хабарламалар тізімін сұрайды. Оны алған кезде ол Көрініске қоңырау шалып, тізімді жібереді және оны веб-бет ретінде көрсетеді. Хабарламаны қосу үшін пайдаланушы пішінді толтырады, содан кейін пішінді деректерді өңдеуге жауапты контроллер шақырылады, ол деректерді қабылдайды және деректер базасына жаңа хабарламаны тексеру және енгізу үшін модельді қолданады, содан кейін HTTP хабарды қарау бетіне қайта бағыттай отырып жауабын береді. Төменде 2.1-суретте көрсетілген.



2.1-сурет – MVC архитектурасы

2.3 Әйелдер киіміне тапсырыс беру бойынша веб-қосымшаны ақпараттық қамтамасыз ету

2.3.1 ДҚБЖ таңдау негіздемесі

Бүгінгі таңда үлкен көлемде деректерді өңдей алатын көптеген ДҚБЖ бар.

Мәліметтер базасын жіктеу келесі критерийлер бойынша жүреді:

- қолдану аясы;
- Инфрақұрылым;
- өңдеу сипаттамалары;
- алынған деректер Саны;

- сақталатын деректерді жіктеу деңгейлері;
- зерттелетін объектілердің сапасы.

Деректердің үлкен көлемін әсер ететін ақпаратқа айналдыра алатын ДҚБЖ жоқ дерекқормен жұмыс істеу мүмкін болмас еді. Үлкен көлемдегі деректерді өңдеу үшін қызметкерге көп уақыт кетеді, сонымен қатар деректерді енгізуде қиындықтар туындауы мүмкін, өйткені қызметкер кейбір маңызды ақпаратты ұмытып кетуі мүмкін.

Деректер базасын дұрыс ұйымдастыру деректерді іріктеудің жылдам және икемді мүмкіндіктерін ұсынады. Бұл іздеу және сұрыптау құралдарын жүзеге асыруды айтарлықтай жеңілдетеді, ал ақпаратқа қол жеткізу құқығының мәселелері көптеген дерекқорды басқару жүйелеріндегі (ДҚБЖ) артықшылықтарды бақылау құралдарының көмегімен шешіледі. Сонымен қатар, деректерді репликациялау және мұрағаттау процестері жеңілдетіледі.

SQL тілі барлық дерлік ДҚБЖ-нің негізі болып табылады. Деректер базасымен жұмыс істеу мысалдарын қарастыруға көшу үшін, кем дегенде, SQL қалай жұмыс істейтінін елестету қажет.

SQL әдетте реляциялық деректер базасымен әрекеттесу үшін қолданылатын стандартты тіл ретінде сипатталады. Алайда, SQL C, C++ немесе PHP сияқты бағдарламалау тілі емес. Керісінше, бұл пайдаланушыға стандартты командалар жиынтығын ұсынатын мәліметтер базасымен әртүрлі операцияларды орындауға арналған интерфейс құралы.

SQLite – бұл дерекқорды басқару жүйесі, оның ерекшелігі оның қосымшаларға енуі болып табылады. Бұл ДҚБЖ-нің көпшілігі өзара әрекеттесу клиент-сервер қағидаты бойынша ұйымдастырылған тәуелсіз қосымшалар екенін білдіреді. Клиент-бағдарлама SQL тілінде сұрау жібереді, ДҚБЖ, оның ішінде қашықтағы компьютерде болуы мүмкін, сұрау нәтижесін қайтарады.

Өз кезегінде, SQLite-бұл бағдарламаға динамикалық немесе статикалық түрде қосылған с тілінде жазылған кітапхана. Көптеген бағдарламалау тілдері үшін SQLite кітапханасына арналған сілтемелер (API) бар. Сонымен, Python-да SQLite ДҚБЖ `Import sqlite3` командасымен импортталады. Сонымен қатар, `sqlite3` модулі стандартты тіл кітапханасына кіреді және бөлек орнатуды қажет етпейді.

Екінші жағынан, SQLite кітапханасын әзірлеуші сайтынан жүктеуге болады. Ол `sqlite3` консольдік утилитасына салынған, оның көмегімен сіз таза SQL-де мәліметтер базасын құра және басқара аласыз. Сондай-ақ, үшінші тарап әзірлеушілерінің графикалық интерфейсі бар SQLite қосымшалары бар.

SQLite ДҚБЖ стандартты SQL-ді қолдайды. SQLite-де қолданылатын SQL тілінің диалектісі синтаксис бойынша PostgreSQL-де қолданылатынға ұқсас. Алайда, SQLite SQL-ге бірқатар ерекшеліктерді қояды.

SQLite-ді ДҚБЖ кітапханасы және мәліметтер базасы ретінде ажырату керек. SQLite көмегімен бір платформалы мәтіндік файл болып табылатын мәліметтер базасы жасалады. Дерекқор файлы, SQLite-тен айырмашылығы, қосымшаға енбейді, оның бөлігі болмайды, ол бөлек болады. Осылайша сіз `sqlite3` консолін қолдана отырып, мәліметтер базасын құра аласыз, содан кейін

оны бағдарламалау тілінің SQLite кітапханасын қолдана отырып Бағдарламада қолдана аласыз. Бұл жағдайда дерекқор файлы жергілікті машинада сақталады.

SQLite қосымшасы оның функционалдығын қарапайым функция қоңыраулары арқылы пайдаланады. Функциялар бағдарлама жұмыс істейтін процесте шақырылғандықтан, қоңыраулар процессаралық өзара әрекеттесу жағдайына қарағанда тезірек жұмыс істейді.

Клиент - серверлік модельден кету SQLite-бұл оқу немесе қысқартылған ДҚБЖ дегенді білдірмейді. Бұл оны ендірілген компонент ретінде қолдану ерекшелігін ғана білдіреді. Шағын жергілікті дерекқорды қажет ететін "ноутбуктерден" бастап браузерлер мен операциялық жүйелерге дейінгі қосымшалардың көптеген түрлері бар.

Біріншіден, бірінші модульдік жүйе жұмысын бастайды. Ол жаһандық қабаттардың жұмысын бастайды, белгілі бір жад көлемін бөледі, конфигурациялық деректер пакеттерін және пәрмен жолдарының параметрлерін оқиды және байланыс менеджеріне бүкіл жұмысты бақылауды аударады.

Клиент өз сұрауын жіберген кезде, ағынды менеджер жұмысқа қосылады, ол клиент үшін жеке ағынды дамытады, содан кейін клиенттің аутентификация процесі жүреді.

Әрі қарай, талдау Модулінің кезегі қолайлы, содан кейін кәштеу процесі жүреді.

Осыдан кейін ақпарат модульдік оңтайландыру немесе көбейту жүйесіне түсе алады. Нәтижесінде барлық ақпарат клиентке қайтарылады.

Реляциялық ДҚБЖ-де деректер өзара байланысты кестелер жиынтығы түрінде ұйымдастырылады. Кестелер арасындағы байланыс басқа кестелердегі мәліметтерге сілтеме түрінде жүзеге асырылады. Кестені екі өлшемді массив ретінде ұсынуға болады, онда әр элементтің орналасуы жол мен бағанның белгілі бір мәндерімен сипатталады.

Үлкен көлемді деректерді өңдеу қажеттілігі базалық есептеу инфрақұрылымына бірегей талаптар қояды. Үлкен көлемді және деректер түрлерін жылдам өңдеу үшін қажетті есептеу қуаты бір серверді немесе серверлер кластерін бұзуы мүмкін. Ұйымдар қажетті жылдамдыққа қол жеткізуде үлкен деректер бар тапсырмаларды шешу үшін барабар есептеу қуатын қолдануы тиіс. Бұл кластерлік архитектураларда өңдеуді және бірлесіп жұмыс істей алатын жүздеген немесе мыңдаған серверлерді талап етуі мүмкін. Экономикалық тиімді тәсілмен мұндай жылдамдыққа қол жеткізу де проблема болып табылады. Кәсіпорындардың, әсіресе тәулік бойы жұмыс істемейтіндердің көптеген басшылары үлкен деректердің жұмыс жүктемелерін қолдау үшін серверлер мен қоймалардың кең инфрақұрылымына қаражат салуға тырысады. Жалпы қол жетімді бұлтты есептеулер нәтижесінде қазіргі уақытта үлкен деректер жүйесін орналастырудың негізгі құралы болып табылады. Жалпы бұлт провайдері 10 деректер петабайларын сақтай алады және үлкен деректерді талдау жобасын орындау үшін серверлердің қажетті санын көбейте алады. Бизнес іс жүзінде қолданылған сақтау және есептеу

уақыты үшін ғана төлейді және бұлттың даналарын қайтадан қажет болғанша өшіруге болады. Қызмет көрсету деңгейін одан әрі арттыру үшін, жалпы қол жетімді бұлт қызметінің жеткізушілері Apache Hadoop жоғары таралған есептеу даналарын, Apache Spark өңдеу механизмін және тиісті үлкен деректер технологияларын қамтитын басқару қызметтері арқылы үлкен деректермен жұмыс істеуге мүмкіндік береді. Amazon Web Services (AWS) компаниясынан Amazon Elastic MapReduce (EMR) - көпшілік бұлтта жұмыс істейтін үлкен деректер қызметінің мысалдарының бірі; басқалары Microsoft Azure HDInsight және Google Cloud Dataproc қамтиды. Бұлт ортасында үлкен деректер таратылған Hadoop (HDFS) файлдық жүйесінде немесе Amazon Simple Storage Service (S3) сияқты арзан бұлт қоймасында сақталуы мүмкін.); NoSQL деректер қоры қолданбаларға арналған бұлтта жақсы нұсқалардың тағы бі түрі болып табылады. Жергілікті үлкен деректер жүйесін өрістетуді қалайтын ұйымдар үшін, әдетте, Hadoop және Spark үшін қосымша болатын ашық бастапқы коды бар Apache технологиясы, оның ішінде ресурстарды келісудің тағы бір құралы (YARN), Hadoop-қа енгізілген ресурстар менеджері және тапсырмаларды жоспарлаушы; MapReduce бағдарламалау шеңбері; Kafka, қосымшалар мен деректерді ағындық жіберу арасында хабар алмасу платформасы; HBase деректер қоры; Drill, HiveИмпала және сол сияқты SQL-on-Hadoop сұраныстарының механизмдері мен тетіктері пайдаланушылар ашық бастапқы коды бар технологиялар нұсқаларын өздері орната алады немесе Cloudera, Hortonworks және MapR Technologies ұсынатын үлкен деректердің коммерциялық платформаларына жүгіне алады, олар да бұлтта қолдауын көрсетеді. Алайда, Cloudera және Hortonworks қазан айында қосылу туралы келісті 2018, бұл қол жетімді жергілікті платформалар санын екіге дейін қысқартуы мүмкін. Үлкен деректерді шағын деректермен, тағы бір дамып келе жатқан терминмен салыстыруға болады, ол мәліметтерді сипаттау үшін жиі пайдаланылады, оның көлемі мен пішімін өзіне-өзі қызмет көрсету талдаушылары үшін оңай пайдалануға болады. Аксиома жиі айтылатын «үлкен деректер машиналарға арналған; шағын деректер адамдарға арналған». Қаржы саласында, әсіресе қаржы қызметтері саласында үлкен деректер, көптеген қосымшаларда пайдаланылады, мысалы: – Қызметкерлердің мониторингі және бақылауы; – Сақтандыру андеррайтерлері сыйақы анықтау үшін және кредит беру туралы шешім қабылдау үшін кредиттік мамандар пайдалана алатындай болжаушы модельдер; – Қаржы нарықтарының бағытын болжау алгоритмдерін әзірлеу;11 – Жылжымайтын мүлік сияқты өтімді емес активтерді бағалау. Қаржы мекемелері үлкен деректерді кеңінен пайдаланады: киберқауіпсіздікті арттырудан бастап клиенттердің кетуін төмендетуге дейін, заманауи банктік қызметтерді жоғары персонализацияланған түрге келтіретін инновациялық және дербестендірілген ұсыныстар есебінен клиенттердің және тағы да басқалардың ниеттестігін дамыту.

Реляциялық деректер базасымен операциялар қалай орындалады? Ол үшін SQL-де SELECT, INSERT, UPDATE және DELETE сияқты жиі

қолданылатын командалардың арнайы жиынтығы бар. Мысалы, егер сізге ID 2003 көмегімен клиенттің электрондық пошта мекенжайын алу қажет болса, келесі SQL пәрменін орындаңыз:

```
SELECT email FROM customers WHERE id = 2003
```

Нәтижесінде команда келесідей көрінеді:

```
SELECT өріс атауы FROM кесте атауы [WHERE шарт]
```

Өрістің атын таңдаңыз аты шарт орналасқан кестеден.

Жақшалар команданың соңғы бөлігі міндетті емес екенін көрсетеді.

Мысалы, клиенттер кестесінен барлық клиенттердің электрондық пошта мекенжайларын алу үшін келесі сұрауды орындаңыз:

```
SELECT email FROM customers
```

Мысалы, пайдаланушы өнім кестесіне жаңа жазба қосады делік. Ең қарапайым қою командасы:

```
INSERT into products VALUES (100, Фффф);
```

Егер бұл деректерді кейінірек жою қажет болса, келесі пәрменді қолданыңыз:

```
DELETE FROM products WHERE prod_id = 100
```

2.3.2 Деректер базасының құрылымын әзірлеу

Деректер қоры SQLite деректер қорын басқару жүйесінде әзірленді. Ол жерде маңызды 2 кестеден тұрады.

Төменде мәліметтер базасы кестелерінің әзірленген құрылымдары берілген:

- order кестесі;
- orderitem кестесі;
- category кестесі;
- product кестесі;

2.1-кесте – Order кестесі

Өріс атауы	Өріс типі	Сипаттама
first_name	CharField	Аты
last_name	CharField	Тегі
email	EmailField	Электронды пошта
address	CharField	Мекен-жайы
postal_code	CharField	Код
city	CharField	Қала
created	DateTimeField	Құрылған күні
updated	DateTimeField	Жаңартылған күні
paid	BooleanField	Төленгені
coupon	ForeignKey	Купон

2.2-кесте – OrderItem кестесі

Өріс атауы	Өріс типі	Сипаттама
order	ForeignKey	Тапсырыс
product	ForeignKey	Продукт
price	DecimalField	Баға
quantity	PositiveIntegerField	Саны

2.3-кесте – Category кестесі

Өріс атауы	Өріс типі	Сипаттама
name	CharField	Атауы
slug	SlugField	Слаг

2.4-кесте – Product кестесі

Өріс атауы	Өріс типі	Сипаттама
name	CharField	Атауы
slug	SlugField	Слаг
description	TextField	Сипаттамасы
category	ForeignKey	Сыртқы кілт
image	ImageField	Сурет
price	DecimalField	Баға
available	BooleanField	Қолжетімділік
created	DateTimeField	Құрылған күні
updated	DateTimeField	Жаңартылған күні

2.4 Әйелдер киімдеріне онлайн тапсырыс беруді әзірлеу және іске асыру барысында қолданылатын технологиялар

Белгілі бір істі бастамас бұрын қолданатын технологияларды таңдап алу жұмыс барысындағы маңызды факторлардың бірі болып табылады. Интернетті пайдалану эволюциясына байланысты веб-сайттар құру технологиялары ақпараттық кеңістіктің негізгі бірлігіне айналды. Веб-ресурстар (сайттар, қосымшалар, сервистер) санының өсу динамикасы сайтты құру технологиясы қол жетімді бола бастағанын көрсетеді.

Веб-сайт құрудың негізгі технологияларының тізімін айтатын болсам, ең алдымен, кез – келген сайт екі бөліктен тұрады, визуалды және бағдарламалық. Frontend әзірлеушілер визуалды компонентте жұмыс істейді, ал backend мамандар бағдарламалық жасақтамада жұмыс істейді. FrontEnd HTML, CSS, JavaScript сияқты технологияларды қолданады. Backend негізгі құралдары – PHP, Python, Ruby және т.б. бағдарламалау тілдері. Ендігі кезекте Frontend пен жағында қолданылатын технологияларға толығырақ тоқталып өтейін.

HTML (HyperText Markup Language — "гипермәтіндік белгілеу тілі") — вебтің ең негізгі құрылыс блогы. Ол веб-мазмұнның мазмұны мен құрылымын анықтайды. HTML-ден басқа технологиялар, әдетте, веб-беттің сыртқы

түрін/көрінісін (CSS) немесе функционалдығын/әрекетін (JavaScript) сипаттау үшін қолданылады. Бірақ HTML-сыз ешқандайда веб-сайт құру мүмкін емес. Гипермәтін ("hypertext") дегеніміз-веб-беттерді бір веб-сайт ішінде немесе веб-сайттар арасында бір-бірімен байланыстыратын сілтемелер. Сілтемелер Вебтің негізгі аспектісі болып табылады. Мазмұнды интернетке жүктеу және оны басқа адамдар жасаған парақтармен байланыстыру арқылы сіз Бүкіләлемдік ғаламтордың белсенді мүшесі боласыз.

HTML веб-шолғышта мәтінді, суреттерді және басқа мазмұнды көрсету үшін арнайы белгілеулерді ("markup") қолданады. HTML белгілеу < head>, < title>, < body>, < header>, < footer>, < article>, < section>, < p>, < div>, < span>, < img>,< side>,< audio>, canvas>, < datalist>, < details>, < embed>, < nav>, < output>, < progress>, <video> және тағы басқалар.

HTML элементі құжаттағы басқа мәтіннен "тегтер" арқылы ерекшеленеді, олар "<" және ">" қоршалған элементтің атауынан тұрады. Тег ішіндегі элемент атауы регистрге сезімтал емес. Яғни, ол жоғарғы немесе төменгі регистрде жазылуы немесе араласуы мүмкін. Мысалы, <title > тегін <Title>, <TITLE > немесе басқа жолмен жазуға болады. Бірақ тек HTML тілін қолданатын болсақ веб-сайтымыз мүлде қарапайым болып қалады. Сондықтан да сайтты кішкене рәсімдеу үшін CSS қолданамыз.

CSS (Cascading Style Sheets) — бұл веб-бетті сәндеу үшін қолданылатын код. HTML сияқты, CSS іс жүзінде бағдарламалау тілі емес. Бұл белгілеу тілі емес – бұл стиль кестесінің тілі. Бұл HTML құжаттарындағы элементтерге стильдерді іріктеп қолдануға мүмкіндік береді дегенді білдіреді. Мысалы, HTML парағындағы абзацтың барлық элементтерін таңдап, олардың ішіндегі мәтінді қара түстен қызылға өзгерту 2.2-суреттегі кодты жазу керек:

```
p {  
  color: red;  
}
```

2.2-сурет – CSS синтаксисі

Суреттегі кодты толығырақ түсіндіріп кететін болсам:

Бүкіл құрылым ережелер жиынтығы деп аталады (бірақ көбінесе қысқа "ереже" үшін). Сондай-ақ, жеке бөліктердің аттарын атап өтемін.

Селектор (Selector)

Ережелер жиынтығының басындағы элементтің HTML атауы. Ол стильді қолдану үшін элементті(лерді) таңдайды (бұл жағдайда р элементтері). Басқа элементті сәндеу үшін селекторды өзгертіңіз.

Хабарландыру (Declaration)

Түс сияқты жалғыз ереже: қызыл; стильдендіргіңіз келетін элементтің қандай қасиеттерін көрсетеді.

Қасиеттері (Properties)

Белгілі бір HTML элементін стильдеудің жолдары (бұл жағдайда түс <р>элементтерінің қасиеті болып табылады). CSS-те сіз өзіңіздің ережеңізде қандай қасиеттерге қол жеткізгіңіз келетінін таңдайсыз.

Қасиет мәні (Property value)

Оң қасиеттері, кейін двоеточия, бізде мәні қасиеттері, ол таңдайды, бірі-көптеген ықтимал белгілері осы қасиеттері (көптеген мәндерін color басқа, red).

Осы қасиеттерді пайдалана отырып веб-сайтымызға стиль береміз. Ал енді бұл сайтқа функционалдық қосу үшін міндетті түрде программалау тілі қажет болады. Жоғарыда айтып өткендей мен бұл программалау тілі ретінде Python программалау тілінің Django фреймворкын таңдадым.

Python – объектіге бағытталған жоғары деңгейлі бағдарламалау тілі.

Бұл тілдің синтаксисі минималистік және қарапайым, стандартты кітапханаларда көптеген пайдалы функциялар бар. Python әртүрлі мәселелерді шешу үшін, соның ішінде веб-қосымшаларды құру үшін қолданылады. Python көмегімен сервер тарапынан HTML кодын динамикалық түрде жасауға болады.

Python негізгі артықшылықтары:

- ыңғайлы және түсінікті бағдарламалық код;
- жүйе модельдерін ыңғайлы және жылдам жасау;
- құжаттаманың үлкен саны, сондай-ақ тез өсіп келе жатқан

Python қолданатын бағдарламашылар қауымдастығы;

- объектіге бағытталған тіл;
- әртүрлі кітапханалардың үлкен саны;
- веб-қосымшаларды құруға арналған әртүрлі құрылымдар.

Python негізгі кемшіліктері:

- ең танымал бағдарламалау тілі емес;
- көп ағынды жетілмегендігі;
- деректер базасымен жұмыс істеу құралдарының бастапқы шектеулілігі, бірақ кейіннен жойылады.

Django – Python программалау тілінде жазылған өте танымал және толық жұмыс істейтін серверлік веб-шеңбер. Бұл модуль Django-ның ең танымал серверлік веб-шеңберлердің бірі, Даму ортасын қалай орнату және оны өзіңіздің веб-қосымшаларыңызды жасау үшін қалай пайдалану керектігі туралы әңгімелейді. Django "барлық қосылған" философиясын ұстанады және

әзірлеушілер "қораптан тыс" жасағысы келетін барлық нәрсені ұсынады. Сізге қажет барлық нәрсе бір "өнімнің" бөлігі болғандықтан, бәрі бірге жұмыс істейді, дизайнның дәйекті қағидаларына сәйкес келеді және кең және өзекті құжаттамаға ие. Django кез-келген веб-сайтты құру үшін пайдаланылуы мүмкін. Мазмұнды басқару жүйелері мен wiki-ден бастап әлеуметтік медиа мен жаңалықтар сайттарына дейін. Ол кез-келген клиенттік ортамен жұмыс істей алады және кез-келген форматта (HTML, RSS арналары, JSON, XML және т.б.) мазмұнды жеткізе алады.

Django сізге қажет кез-келген функционалдылыққа арналған шешімдерді ұсынады (мысалы, бірнеше танымал мәліметтер базасы, шаблондар және т.б.), сонымен қатар қажет болған жағдайда оны үшінші тарап компоненттері кеңейте алады.

Django программисттерге сайтты автоматты түрде қорғау үшін "дұрыс нәрсені жасау" үшін жасалған шеңберді ұсыну арқылы көптеген жалпы қауіпсіздік қателіктерін болдырмауға көмектеседі. Мысалы, Django пайдаланушының есептік жазбалары мен құпия сөздерін басқарудың қауіпсіз әдісін ұсынады, мысалы, сеанс туралы ақпаратты осал жерде cookie файлдарына орналастыру (оның орнына cookie файлдары тек кілтті қамтиды және нақты деректер дерекқорда сақталады) немесе пароль хэшінің орнына құпия сөздерді тікелей сақтау.

Пароль хэші – бұл криптографиялық хэш функциясы арқылы парольді өңдеу арқылы жасалған бекітілген ұзындық мәні. Django енгізілген парольді хэш функциясы арқылы өткізіп, нәтижені сақталған хэш мәнімен салыстыру арқылы тексере алады. Функцияның "бір жақты" сипатына байланысты, сақталған хэш мәні бұзылған болса да, шабуылдаушыға бастапқы парольді анықтау қиынға соғады.

Django әдепкі бойынша көптеген осалдықтардан қорғайды, соның ішінде SQL инъекциясы, сайт аралық сценарий, сайт аралық сұраулар және кликджекинг.

Сондай-ақ, компоненттік "shared-nothing" архитектурасын қолданады (оның әр бөлігі басқаларына тәуелсіз, сондықтан қажет болған жағдайда ауыстырылуы немесе өзгертілуі мүмкін). Бөлшектерді нақты бөлу дегеніміз, Django кез-келген деңгейде жабдықты қосу арқылы трафикті көбейте алады: кэштеу серверлері, дерекқор серверлері немесе қосымшалар серверлері. Ең көп жүктелген сайттардың бірі Django-ны сәтті масштабтады (мысалы, Instagram және Disqus, олардың екеуін ғана атаңыз).

Django коды қолдау көрсетілетін және қайта пайдаланылатын кодты құруға көмектесетін принциптер мен дизайн шаблондарын қолдана отырып жазылған. Атап айтқанда, ол "Don't Repeat Yourself" (DRY, "қайталамаңыз") принципін қолданады, сондықтан қажет емес қайталану жоқ, бұл код көлемін азайтады. Django сонымен қатар байланысты функцияларды қайта пайдаланылатын "қосымшаларға" топтастыруға ықпал етеді және төменгі деңгейде байланысты кодты модульдерге топтайды (Model View Controller (MVC) үлгісіне сәйкес).

Джанго көптеген платформаларда жұмыс істейтін Python-да жазылған. Сонымен қатар, Django-ны көптеген веб-хосттар жақсы қолдайды, олар көбінесе Django сайттарын орналастыру үшін белгілі бір инфрақұрылым мен құжаттаманы ұсынады.

Django 2003 жылдан 2005 жылға дейін газет веб-сайттарын құруға және қызмет көрсетуге қатысқан топпен жасалды. Бірнеше веб-сайт жасағаннан кейін команда көптеген жалпы кодтар мен дизайн үлгілерін қайта қолдана бастады. Бұл жалпы код 2005 жылдың шілдесінде "Django" ашық бастапқы жобасына айналған веб-шеңберге айналды.

Джанго өзінің алғашқы шығарылымынан (1.0) 2008 жылдың қыркүйегінде жаңадан шыққан 3.1 (2020) нұсқасына дейін өсіп, жетілдірілуде. Әр нұсқада жаңа функциялар қосылады және қателер түзетіледі, жаңа мәліметтер базасын, шаблондарды және кэштеуді қолдаудан бастап, "жалпы" көру функциялары мен сыныптарын қосуға дейін (әзірлеушілер бірқатар бағдарламалық тапсырмалар үшін жазуы керек код көлемін азайту).

Django – бұл мыңдаған пайдаланушылар мен мүшелер жұмыс істейтін өркендеп келе жатқан ашық бастапқы жоба. Оның шығу тегін көрсететін кейбір ерекшеліктері бар болса да, джанго кез-келген веб-сайтты жасай алатын әмбебап фреймворкка айналды.

Django қолданатын үлкен сайттардың санына, қатысушылардың санына және ақысыз және ақылы қолдауды ұсынатын адамдардың санына сүйене отырып, жауап беруге болады: Иә, Django-танымал фреймворк!

Джангоны Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic, Open Knowledge Foundation, Pinterest және ашық стек сияқты ірі сайттар пайдаланады.

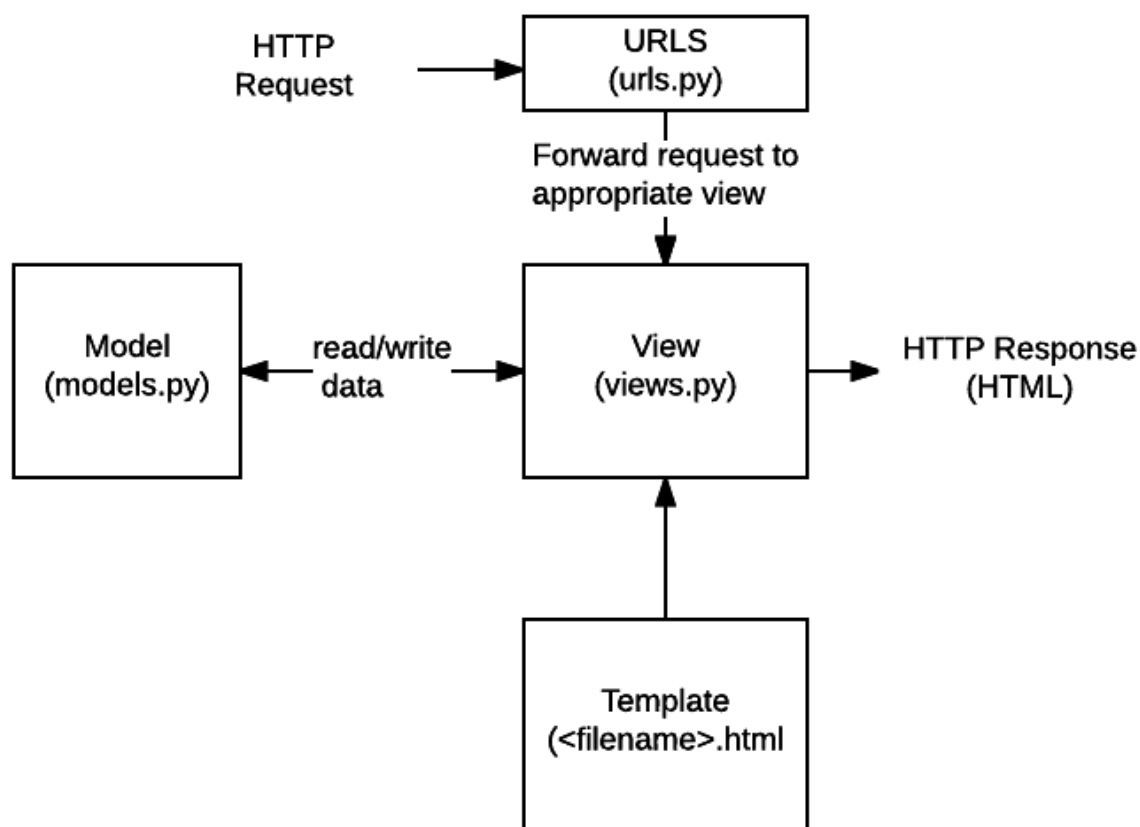
Веб-фреймворктарды көбінесе "икемді" және "икемсіз" деп бөлуге болады.

Икемсіз – бұл белгілі бір мәселені шешудің "дұрыс жолы" бар адамдар. Олар көбінесе белгілі бір аймақта жылдам орналастыруды қолдайды (белгілі бір типтегі мәселелерді шешу), өйткені кез-келген нәрсені жасаудың дұрыс әдісі әдетте жақсы түсініледі және жақсы құжатталады. Алайда, олар негізгі саладан тыс мәселелерді шешуде икемді болмауы мүмкін және олар қандай компоненттер мен тәсілдерді қолдана алатындығының аз нұсқаларын ұсынады.

Керісінше, икемді фреймворктарда мақсатқа жету үшін немесе қандай компоненттерді қолдану керек болса да, компоненттерді байланыстырудың жақсы әдісіне шектеулер аз болады. Олар әзірлеушілерге белгілі бір тапсырманы орындау үшін ең қолайлы құралдарды қолдануды жеңілдетеді, дегенмен бұл компоненттерді өзіңіз табуыңыз керек.

Джанго "орташа икемді", сондықтан "екі әлемнің ең жақсысын" қамтамасыз етеді. Ол көптеген веб-әзірлеу тапсырмаларын өңдеуге арналған компоненттер жиынтығын және оларды қолданудың бір (немесе екі) әдісін ұсынады. Алайда, мұндай Django архитектурасы сіз әдетте бірнеше түрлі опцияларды таңдай аласыз немесе қажет болған жағдайда жаңа нұсқаларға қолдау қоса аласыз дегенді білдіреді.

Дәстүрлі ақпараттық веб-сайтта веб-қосымша веб-шолғыштан (немесе басқа клиенттен) HTTP сұрауларын күтеді. Сұрау алынған кезде, бағдарлама URL мекен-жайы және POST немесе GET сұрауларындағы мәліметтер негізінде қажет нәрсені жасайды. Не талап етілетініне байланысты, ол мәліметтер базасынан ақпаратты оқи немесе жаза алады немесе сұранысты қанағаттандыру үшін қажет басқа тапсырмаларды орындай алады. Содан кейін бағдарлама жауапты веб-шолғышқа қайтарады, көбінесе HTML шаблонна алынған деректерді енгізу арқылы шолғышта көрсетілетін HTML бетін динамикалық түрде жасайды. Django-да жазылған Веб-қосымшалар, әдетте, осы қадамдардың әрқайсысын жеке файлдарға өңдейтін кодты топтайды.



2.3-сурет – Django фреймворкының құрылымы

URLs: әр URL мекен-жайынан сұрауларды бір функциямен өңдеуге болатынына қарамастан, әр ресурсты өңдеу үшін бөлек функция жазу әлдеқайда ыңғайлы. URL маршрутизаторы HTTP сұрауларын сұрау URL мекен-жайы негізінде тиісті көрініске бағыттау үшін қолданылады. Сонымен қатар, URL маршрутизаторы берілген шаблонға сәйкес URL мекен-жайынан деректерді шығарып, оларды дәлел ретінде тиісті дисплей функциясына (view) жібере алады.

View: view (ағылш. "көрсетілім") – HTTP сұрауларын алатын және жауаптарды қайтаратын сұрау өңдегішінің функциясы. View функциясы

сұраныстарды қанағаттандыру үшін қажетті деректерге қол жеткізе алады және жауаптарды үлгілер арқылы шаблондарға жібереді.

Models: Модельдер – бұл Python нысандары, олар қосымшаның деректер құрылымын анықтайды және Дерекқорға сұраныстарды басқару (қосу, өзгерту, жою) және орындау тетіктерін ұсынады.

Templates: Template (ағылш. "шаблон") – бұл беттің құрылымын немесе орналасуын анықтайтын мәтіндік файл (мысалы, HTML беттері), ағымдағы мазмұнды шығару үшін қолданылатын алмастыру өрістері бар. Көрініс HTML шаблондарын қолдана отырып және оларды модельден (model) мәліметтермен толтыру арқылы HTML беттерін динамикалық түрде жасай алады. Шаблонды HTML емес, кез-келген типтегі файл құрылымын анықтау үшін пайдалануға болады.

2.5 UML модельдеу тілі

Бірыңғай модельдеу тілі (UML) бағдарламалық қамтамасыздандырудың «сызбаларын» құрудың стандартты құралы болып табылады. Оны бағдарламалық жасақтама жүйесінің артефактілерін визуализациялау, көрсету, жобалау және құжаттау үшін UML пайдалана аламыз.

UML кез-келген жүйені модельдеуге жарамды: кәсіпорынның ақпараттық жүйелерінен бастап таралған веб-қосымшаларға дейін және тіпті нақты уақыттағы жүйелерге енгізілген. UML - бағдарламалық жасақтама жүйесінің артефактілерін визуализациялау, нақтылау, құрастыру және құжаттау тілі. UML-ге ұқсас модельдеу тілі - «сызбаларды» бағдарламалық жасақтаманың стандартты құралы.

Кез-келген жүйені түсіну үшін біз өзара байланысты көптеген модельдерді жасауымыз керек. Бағдарламалық жасақтамаға қолданған кезде, бұл оның даму циклі барысында әр түрлі тұрғыдан жүйелік архитектураның сипаттамаларын сипаттауға болатын тіл қажет дегенді білдіреді.

UML - бұл бағдарламалық жасақтаманы әзірлеу және орналастыру кезінде ескерілуі керек талдау, дизайн және іске асырумен байланысты дәл және толық графикалық модельдерді құруды білдіретін графикалық спецификация тілі.

UML - бұл дизайн тілі, оны қолдану арқылы жасалған модельдер әртүрлі бағдарламалау тілдеріне тікелей аударыла алады. Басқаша айтқанда, UML моделін Java, C ++, Visual Basic сияқты тілдермен салыстыруға болады және тіпті дерекқордың реляциялық кестелерін бірге қоса.

Бұл модельді бағдарламалау тілімен салыстыру тікелей дизайн жасауға мүмкіндік береді: UML моделінен белгілі бір тілде код құру. Кері есепті де шешуге болады: қолданыстағы енгізу негізінде модельді қайта құру.

UML жүйенің архитектурасын және оның барлық бөлшектерін құжаттандыру мәселесін шешеді, жүйеге қажеттіліктер мен тесттерді анықтауға арналған тіл ұсынады, ақырында жобаны жоспарлау және нұсқаны басқару кезеңінде модельдеу үшін құрал ұсынады.

UML-дың тиімді пайдаланылуы:

- кәсіпорынның ақпараттық жүйелері;
- банктік және қаржылық қызметтер;
- телекоммуникациялар;
- көлікте;
- қорғаныс өнеркәсібі, авиация және ғарыш;
- сауда;
- медициналық электроника;
- ғылымда;
- веб-жүйелерде.

UML қолдану аясы бағдарламаны модельдеуімен шектелмейді. Ол, мысалы, құқықтық жүйелердегі құжат айналымын, ауруханалардағы пациенттерге күтім жасау жүйесінің құрылымы мен жұмыс істеуін, конструкторлық жабдықтарды модельдеуге мүмкіндік береді.

UML тілін конструктивті қолдану күрделі жүйелерді модельдеудің жалпы принциптерін және объектіге бағытталған талдау мен дизайн (ООАР) процесінің ерекшеліктерін түсінуге негізделген. Күрделі жүйелердің модельдерін құруға арналған экспрессивті құралдарды таңдау бірнеше принциптерге негізделген.

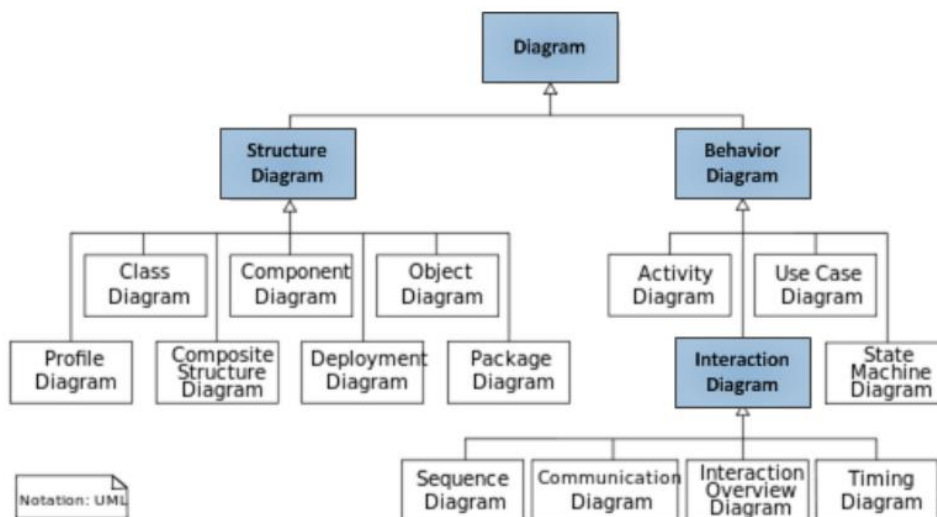
Біріншісі - абстракция қағидасы, модельге тек жүйенің функцияларының орындалуымен немесе оның мақсатына тікелей байланысты аспектілерді ғана қосуды ұйғарады. Бұл жағдайда алынған үлгіні талдау мен зерттеу процесін қиындатпау үшін барлық ұсақ бөлшектер алынып тасталады.

Күрделі жүйелер модельдерін құрудың екінші ұстанымы - көп модельдеу принципі. Бұл күрделі жүйенің бірде-бір көрінісі оның барлық мүмкіндіктерін жеткілікті түрде білдіруге жеткіліксіз екенін білдіреді.

Қолданбалы жүйелік талдаудың тағы бір принципі күрделі жүйелер модельдерінің иерархиялық құрылысы принципі болып табылады. Бұл принцип абстракцияның немесе деңгейлердің әртүрлі деңгейлерінде модельді құру процесін қарастыруды талап етеді.

Осылайша, ООАР процесі жалпы модельдер мен тұжырымдамалық деңгей ұсынуларынан логикалық және физикалық деңгейдің нақтыланған және егжей-тегжейлі көріністеріне дейін деңгей бойынша бөлінуі мүмкін. ООАР-тың әр кезеңінде бұл модельдер толықтырылып отырады, бұл оларға күрделі жүйені нақты іске асырудың әртүрлі аспектілерін неғұрлым дәл көрсетуге мүмкіндік береді.

Нысанға бағытталған талдау және жүйелік дизайн UML сөздігін пайдалануды қамтиды, оның құрамына үш типті блоктар кіреді: нысандар, қатынастар және диаграммалар.

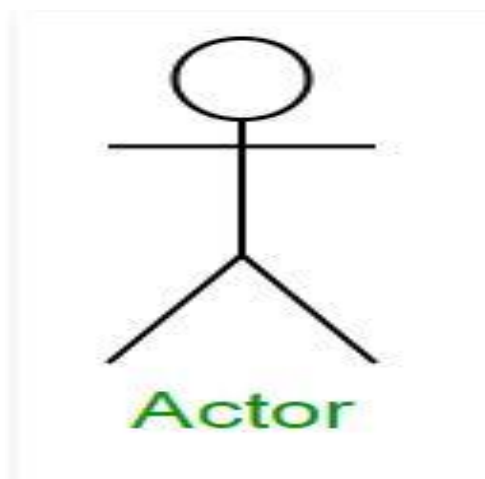


2.4-сурет – Диаграммаларға шолу

2.6 Тізбек диаграммасы

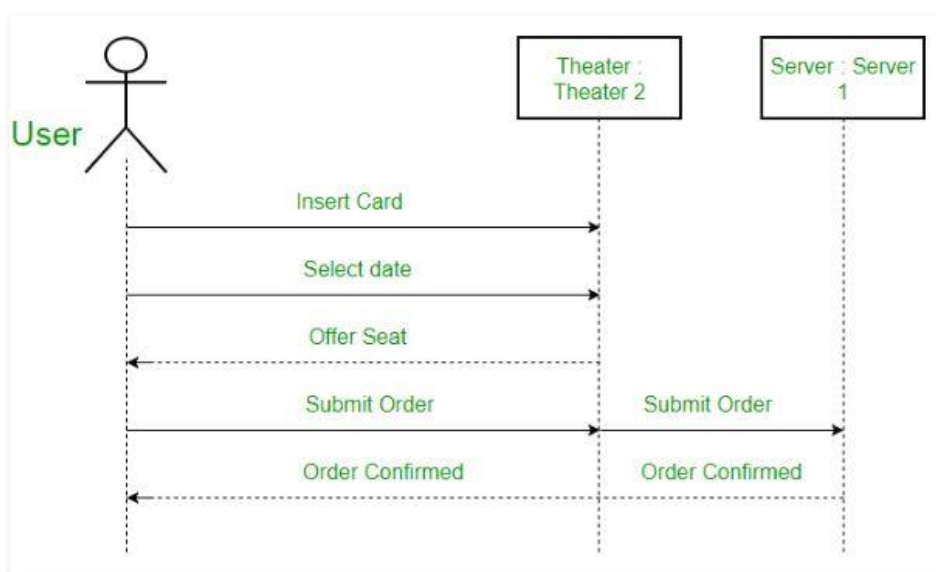
UML тізбек диаграммалары - бұл әрекеттерді қалай жүзеге асыратындығын көрсететін өзара әрекеттесу диаграммалары. Олар ынтымақтастық шеңберіндегі нысандардың өзара әрекетін түсіреді. Тізбек диаграммалары уақыттың фокустары болып табылады және олар хабарлардың қай уақытта және қай уақытта жіберілетінін көрсету үшін диаграмманың тік осін пайдаланып көрнекіліктің ретін көрсетеді. Тізбектік диаграммасы жай объектілердің өзара әрекеттесуін тізбектелген тәртіппен, яғни осы өзара әрекеттесулер болатын ретпен бейнелейді. Реттеу диаграммасына сілтеме жасау үшін оқиға оқиғалары мен сценарийлер терминдерін де қолдана аламыз. Бірізділік диаграммалары жүйелік функциядағы объектілерді қалай және қандай ретпен сипаттайды. Бұл диаграммаларды жаңа және қолданыстағы жүйелерге қойылатын талаптарды түсіну және түсіну үшін кәсіпкерлер мен бағдарламалық жасақтама жасаушылар кеңінен қолданады. Тізбек диаграммасы туралы жазбалар:

Актерлер - UML диаграммасындағы актер жүйе мен оның объектілерімен өзара әрекеттесетін рөл түрін білдіреді. Бұл жерде актер әрдайым UML диаграммасын қолдана отырып, модельдейтін жүйенің шеңберінен тыс болатындығын ескерген жөн.



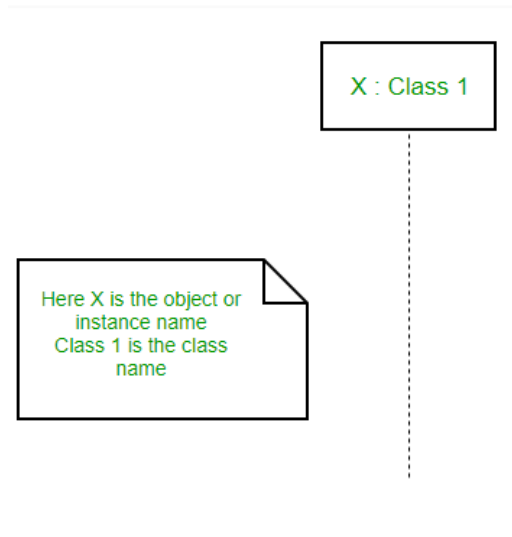
2.5-сурет – Актердың белгіленуі

Біз әр түрлі рөлдерді бейнелеу үшін актерлерді қолданамыз, соның ішінде адам қолданушылары және басқа сыртқы тақырыптар. Біз UML диаграммасында актерлерді жеке адамның жазбасын қолдана отырып ұсынамыз. Тізбек диаграммасында бірнеше актер бола аламыз. Мысалы, мұнда орынды брондау жүйесіндегі пайдаланушы жүйеден тыс жерде және жүйенің бөлігі болып табылмайтын жерде актер ретінде көрсетіледі.



2.6-сурет – орындарды брондау жүйесімен әрекеттесетін актер

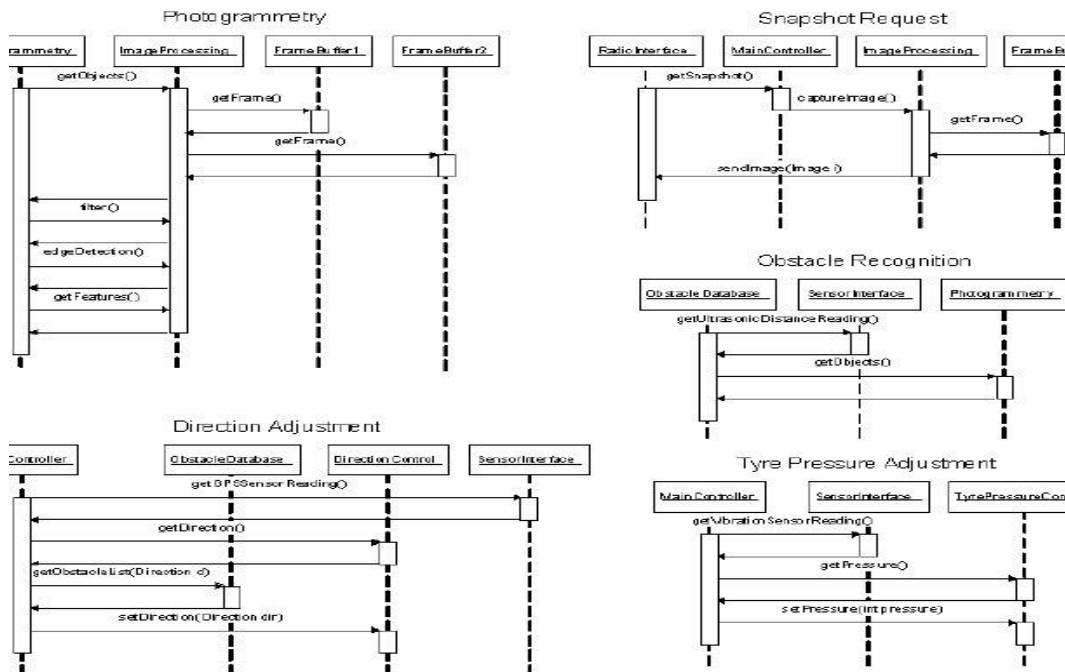
Өмірлік сызық - бұл тізбектік диаграммадағы жеке қатысушыны бейнелейтін аталатын элемент. Осылайша, кезек-кезек диаграммасындағы әр инстанция өмір сызығымен ұсынылған. Сызықтық элементтер жоғарғы жағында тізбектік диаграммада орналасқан. UML-дегі өмір сызығын атауға арналған стандарт келесі форматқа сәйкес келеді.



2.7-сурет – Өмірлік сызық сипаттамасы

Біз өмір сызығын басы деп аталатын тіктөртбұрышта оның аты мен түрімен көрсетеміз. Басы жоғарыда көрсетілгендей тік сызық сызықтың жоғарғы жағында орналасқан (баған деп аталады). Егер біз аты аталмаған дананы модельдегіміз келсе, біз сол үлгі бойынша жүреміз, тек қазір өмір жолының аты бос қалдырылады.

Өмір сызығы мен актер арасындағы айырмашылық - өмір сызығы әрқашан жүйенің ішіндегі затты бейнелейді, ал актерлер жүйеден тыс нысандарды бейнелеу үшін қолданылады. Төменде тізбектік сызбаның мысалы келтірілген:

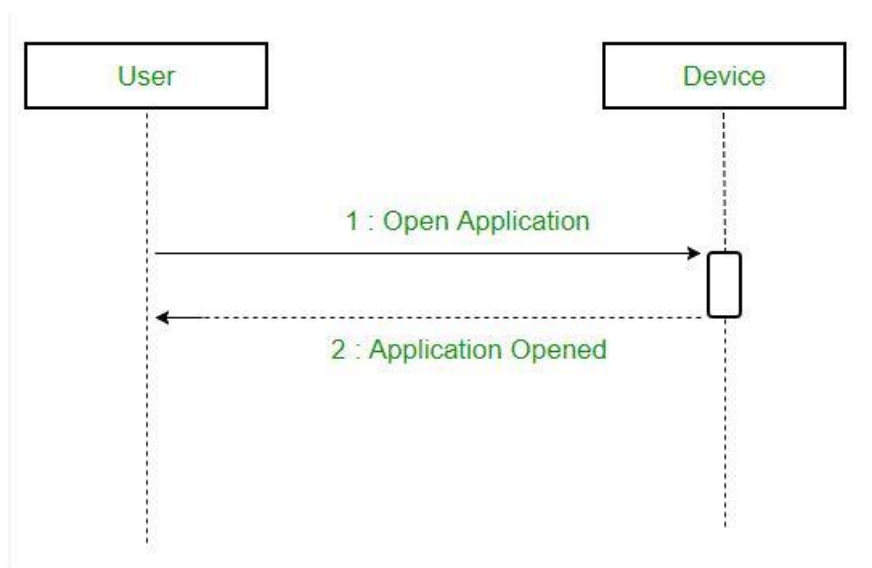


2.8-сурет – Тізбек диаграммасы

Хабарлар - нысандар арасындағы байланыс хабарламаларды қолдану арқылы бейнеленеді. Хабарламалар өмір жолында бірізді түрде шығады. Біз хабарларды көрсеткілерді қолдана отырып ұсынамыз. СЫЗЫҚТЫҚ СЫЗЫҚТАР МЕН хабарламалар бірізділік диаграммасының өзегін құрайды.

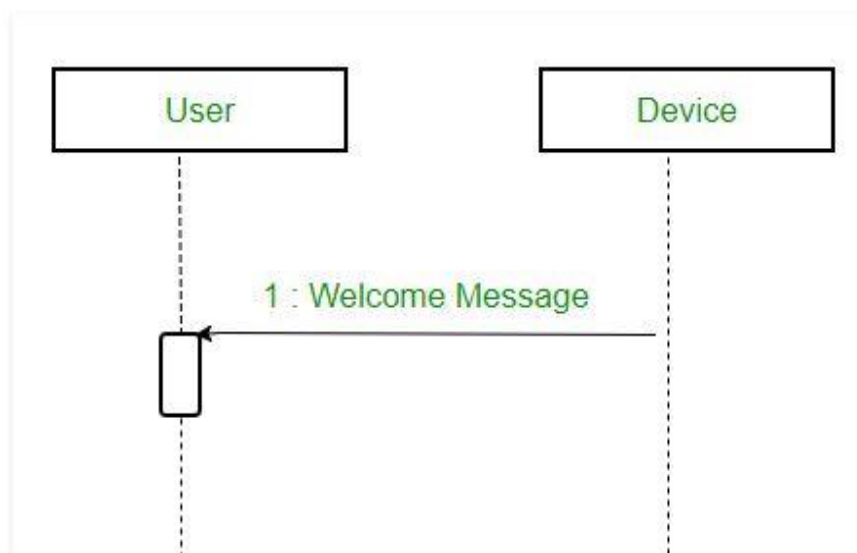
Хабарламаларды келесі санаттарға кеңінен жіктеуге болады:

Синхронды хабарламалар. Синхронды хабарлама өзара әрекеттесу алға жылжудың алдында жауап күтеді. Жіберуші алушы хабарламаны өңдеуді аяқтағанша күтеді. Қоңырау шалушы алушының алдыңғы хабарламаны, яғни жауап хабарламасын алғанын білген кезде ғана жалғасады. Объективті бағдарламалаудағы қоңыраулардың көп саны синхронды. Біз синхронды хабарламаны ұсыну үшін қатты көрсеткі басын қолданамыз.



2.9-сурет – синхронды хабарламалар тізбегінің диаграммасы

Асинхронды хабарлар – асинхронды хабар қабылдағыштан жауап күтпейді. Өзара әрекеттесу алдыңғы хабарламаны өңдейтін немесе қабылдамайтын қабылдағышқа қарамастан алға жылжиды. Асинхронды хабарламаны ұсыну үшін біз бағытталған жебенің басын қолданамыз.



2.10-сурет – асинхронды хабарламалар тізбегінің диаграммасы

Хабарлама жасау - біз дәйектілік диаграммасында жаңа нысанды енгізу үшін Create хабарламасын қолданамыз. Белгілі бір хабарлама қоңырауы объект жасауды қажет ететін жағдайлар болады. Ол нүктелік көрсеткімен көрсетілген және оның Хабарламаны құру символы екенін көрсету үшін оған белгіленген сөз жасаңыз. Мысалы, электрондық коммерция веб-сайтында жаңа тапсырысты құру Тапсырыс класының жаңа объектісін жасауды қажет етеді.

Хабарламаны жою - нысанды жою үшін Жою хабарламасын қолданамыз. Нысан жадқа бөлінген немесе жүйеде жойылған кезде біз хабарламаны жою таңбасын қолданамыз. Бұл жүйеде объектінің пайда болуын жояды. Бұл хпен аяқталатын көрсеткі арқылы ұсынылады.

Өздігінен хабарлама - объект өзіне хабарлама жіберу керек болған кезде белгілі бір сценарий туындауы мүмкін. Мұндай хабарламалар «Өздігінен хабарлар» деп аталады және U пішінді көрсеткімен ұсынылады.

Хабарламаға жауап беру - жауап хабарлары қабылдағыштан жіберушіге жіберілген хабарламаны көрсету үшін қолданылады. Біз нүкте сызығы бар ашық стрелка көмегімен қайтару / жауап хабарламасын ұсынамыз. Өзара әрекеттесу жауап алушы жіберген кезде ғана алға жылжиды.

Табылған хабар - табылған хабар белгісіз дереккөз хабарлама жіберетін сценарийді көрсету үшін қолданылады. Ол аяғынан бастап өмір сызығына бағытталған көрсеткі арқылы ұсынылған. Мысалы, жабдықтың істен шығуы сценарийін қарастыру. Бұл бірнеше себептерге байланысты болуы мүмкін және біз аппараттық құралдың ақауына не себеп болғанын білмейміз.

Жоғалған хабар - жоғалған хабар алушы жүйеге беймәлім сценарийді көрсету үшін қолданылады. Ол өмір сызығынан соңғы нүктеге бағытталған жебе арқылы ұсынылған. Мысалы, ескерту жасалған сценарийді қарастыру. Ескерту пайдаланушыға немесе басқа бағдарламалық жасақтама / объект үшін өмірлік сызық өзара әрекеттесетін болуы мүмкін. Баратын жер алдын-ала белгісіз болғандықтан, біз Жоғалған хабарлама таңбасын қолданамыз.

Сақшылар - жағдайларды модельдеу үшін біз UML-де сақшыларды қолданамыз. Олар хабарлама ағынын шектеу керек деген сылтаумен шектеу қажет болған кезде қолданылады. Сақшылар бағдарламалық жасақтама жасаушыларға жүйеге немесе белгілі бір процеске қатысты шектеулер туралы хабарлау үшін маңызды рөл атқарады.

Тізбек диаграммаларды қолдану:

- Күрделі функцияның, операцияның немесе процедураның артындағы логиканы модельдеу және бейнелеу үшін қолданылады.
- Олар сонымен қатар UML пайдалану жағдайларының диаграммаларын көрсету үшін қолданылады.
- Ағымдағы немесе болашақ жүйелердің толық функционалдығын түсіну үшін қолданылады.
- Хабарламалар мен тапсырмалар жүйеде объектілер немесе компоненттер арасында қалай қозғалатынын елестетіңіз.

2.7 Прецеденттер диаграммасы

Прецеденттер диаграммасы қарапайым болып пайдаланушының жүйемен өзара әрекеттесуінің көрінісі болып табылады, ол қолданушы мен қолданушы қатысатын әр түрлі пайдалану жағдайларын көрсетеді. Прецеденттер диаграммасы жүйені пайдаланушылардың әр түрлі типтерін және әр түрлі пайдалану жағдайларын анықтай алады және көбінесе басқа диаграммалармен бірге жүреді. Прецеденттер шеңберлермен немесе эллипстермен ұсынылған. Прецеденттер диаграммасы - UML-дегі динамикалық немесе мінез-құлық диаграммасы. Активтерді және регистрлерді қолдана отырып, жүйенің функционалдығын модель мысалдарын қолданыңыз. Прецеденттер - бұл жүйе жасауы керек әрекеттер, қызметтер мен функциялар жиынтығы. Бұл тұрғыда «жүйе» дегеніміз - веб-сайт сияқты өнделетін немесе басқарылатын нәрсе. «Актерлер» - бұл жүйе ішінде белгілі рөлдер бойынша жұмыс жасайтын адамдар немесе ұйымдар. Прецеденттер диаграммалары жүйенің функционалдық қажеттіліктерін визуализациялау үшін маңызды болып табылады, бұл дизайнды және дамудың басымдылығын өзгертеді. Олар сонымен қатар жүйеге әсер етуі мүмкін кез-келген ішкі немесе сыртқы факторларды анықтауға көмектеседі және ескеру қажет. Олар жүйеден тыс жоғары сапалы талдауды қамтамасыз етеді. Пайдалану жағдайлары диаграммасы жүйенің осы функцияның қалай іске асырылатындығы туралы толғандырмай, актерлермен қалай әрекеттесетінін көрсетеді.

Прецеденттердің өзі барлық мүмкіндіктер туралы егжей-тегжейлі егжей-тегжейлі егжей-тегжейлі мәлімдей алады, алайда прецеденттер диаграммасы жүйенің жоғары деңгейінің көрінісін қамтамасыз етеді. Бұған дейін «Кейсызбаларды қолдану - бұл сіздің жүйеңіздің сызбасы» деп айтылған. Олар жүйенің іс жүзінде не істеуі керек екенін қарапайым және графикалық түрде ұсынады.

Олардың қарапайым сипатына байланысты кейс-схемаларды қолдану мүдделі тараптар үшін жақсы байланыс құралы бола алады. Сызбалар нақты әлемге еліктеуге тырысады және мүдделі тарапқа жүйенің қалай жасалатынын түсінуге мүмкіндік береді. Шиа мен Ли прецеденттер диаграммаларын қолдануға жарамды жағдайдың бар-жоғын немесе олардың қажет еместігін анықтау үшін зерттеулер жүргізді. Пайдалану жағдайларының диаграммалары жүйенің ниетін мүдделі тараптарға неғұрлым жеңілдетілген түрде жеткізетіндігі және олар «класс диаграммаларынан гөрі толықтай түсіндірілгені» болды.

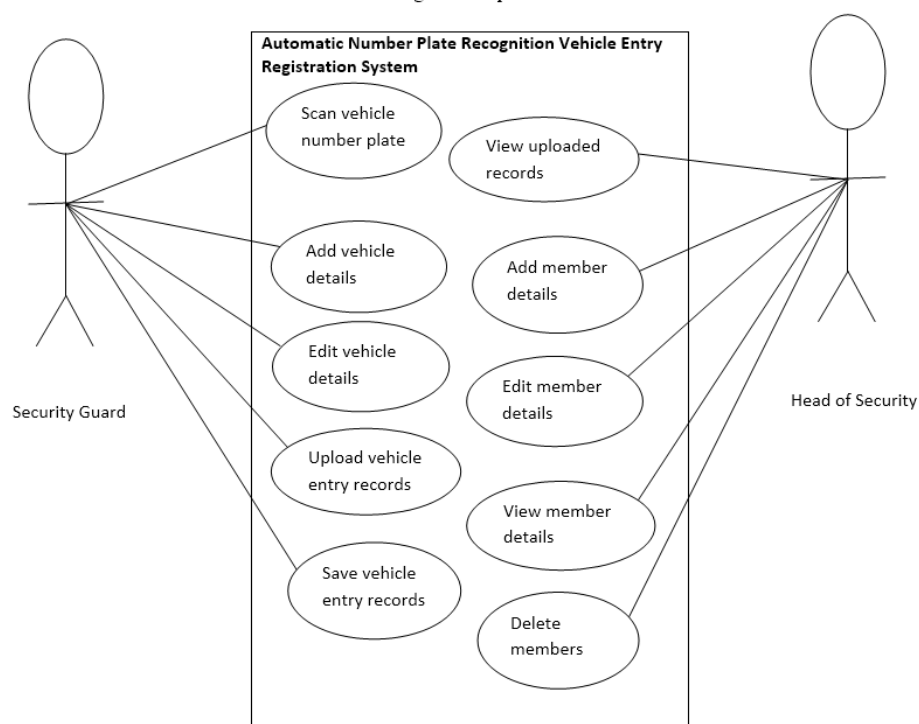
Схемаларды қолдану мақсаты - жүйенің жоғары деңгейде көрінісін қамтамасыз ету және қызығушылық танытқан адамдарға талаптарды қарапайым адамдармен жеткізу. Қосымша диаграммалар мен құжаттаманы жүйенің толық функционалды және техникалық көрінісін қамтамасыз ету үшін пайдалануға болады.

Прецеденттер диаграммасы көптеген егжей-тегжейлерге енбейді. Мысалы, оның қадамдар орындалу ретін модельдейтінін күтпеңіз. Оның орнына пайдалану жағдайларының дұрыс диаграммасы пайдалану жағдайлары, актерлер және жүйелер арасындағы өзара байланыстың жоғары деңгейлі шолуын бейнелейді. Сарапшылар мәтіндік сипаттаманы қолдану жағдайын толықтыру үшін кейс-сызбаларды қолдануды ұсынады.

UML - диаграммаларды құру үшін қолдануға болатын модельдеу құралы. Прецедент таңбаланған сопақ пішінді көрсетілген Stick фигуралары процесстегі актерлерді білдіреді, ал актерлердің жүйеге қатысуы актер мен пайдалану корпусының арасындағы сызықпен модельденеді. Жүйенің шекарасын бейнелеу үшін қолдану қорабына қорап салыңыз.

UML прецеденттер диаграммалары келесі операцияларды орындауға өте қолайлы:

- Жүйені қолданушылардың өзара әрекеттесу мақсаттарын көрсету;
- Жүйеде функционалдық талаптарды анықтау және ұйымдастыру;
- Жүйенің мазмұны мен талаптарын көрсету;
- Пайдалану жағдайындағы оқиғалардың негізгі ағынын модельдеу;



2.11-сурет – преценттер диаграммасы

Преценттер диаграммасының компоненттері:

- Актерлер: жүйемен өзара әрекеттесетін қолданушылар. Актер адам, ұйым немесе сіздің қосымшамен немесе жүйеңізбен өзара әрекеттесетін сыртқы жүйе бола алады. Олар деректерді шығаратын немесе тұтынатын сыртқы нысандар болуы керек.
- Жүйе: Актерлер мен жүйенің өзара әрекеттесуінің белгілі бір реттілігі. Жүйені сценарий деп те атауға болады.

Мақсаты: көп жағдайда қолданудың соңғы нәтижесі. Табысты диаграмма мақсатқа жету үшін қолданылатын әрекеттер мен нұсқаларды сипаттауы керек.

2.8 Класстар диаграммасы

Бағдарламалық жасақтамада бірыңғай модельдеу тіліндегі (UML) классикалық диаграмма - бұл жүйенің кластарын, олардың атрибуттарын, операцияларын (немесе әдістерін) және объектілер арасындағы қатынасты көрсете отырып, жүйенің құрылымын сипаттайтын статикалық құрылым диаграммасы.

Класс диаграммасы объектіге бағытталған модельдеудің негізгі құрылыс блогы болып табылады. Ол қосымшаның құрылымын жалпы тұжырымдамалық модельдеу үшін және модельдерді бағдарламалық кодқа аударатын егжей-тегжейлі модельдеу үшін қолданылады. Класс диаграммаларын мәліметтерді модельдеу үшін де қолдануға болады. Класс диаграммасындағы сабақтар

негізгі элементтерді, қосымшадағы өзара әрекеттестікті және бағдарламаланатын сыныптарды білдіреді.

Диаграммада сабақтар үш бөліктен тұратын қораптармен берілген:

- Жоғарғы бөлікте сынып атауы жазылған. Қалың және ортасында басылады, ал бірінші әріп бас әріппен жазылады.
- Ортаңғы бөлікте сыныптың атрибуттары бар. Олар сол жаққа тураланған және бірінші әріп кіші әріппен жазылған.
- Төменгі бөлікте сынып орындай алатын операциялар бар. Олар сонымен қатар сол жақта орналасқан және бірінші әріп кіші әріппен жазылған.

Жүйені жобалауда бірқатар класстар анықталып, сынып диаграммасында топтастырылған, бұл олардың арасындағы статикалық қатынастарды анықтауға көмектеседі. Егжей-тегжейлі модельдеумен қатар, тұжырымдамалық дизайн сыныптары көбінесе бірнеше ішкі сыныптарға бөлінеді. Жүйелердің әрекетін одан әрі сипаттау үшін осы класс диаграммаларын күй диаграмма немесе UML күй машинасымен толықтыруға болады.

UML атрибуттар мен әдістер сияқты сынып мүшелерін ұсыну механизмдерін және олар туралы конструкторлар сияқты қосымша ақпаратты ұсынады. Класс мүшесінің көрінуін анықтау үшін (яғни, кез-келген атрибут немесе әдіс), бұл жазбалар мүшенің атының алдына қойылуы керек:

- + Public
- Private
- # Protected
- ~ Package

Класс диаграммасының мақсаты:

- Жүйеде классификаторлардың статикалық құрылымын көрсетеді;
- Диаграмма UML тағайындаған басқа құрылымдық диаграммалар үшін негізгі жазбаны ұсынады;
- Әзірлеушілерге және команданың басқа мүшелеріне де пайдалы;
- Бизнес-аналитиктер жүйелерді бизнес тұрғысынан модельдеу үшін сынып диаграммаларын қолдана алады;

UML класс диаграммасы:

- Класстар жиынтығы және
- Класстар арасындағы қатынастар жиынтығы

Жүйеде ұқсас рөлдері бар объектілер тобының сипаттамасы, олар мыналардан тұрады:

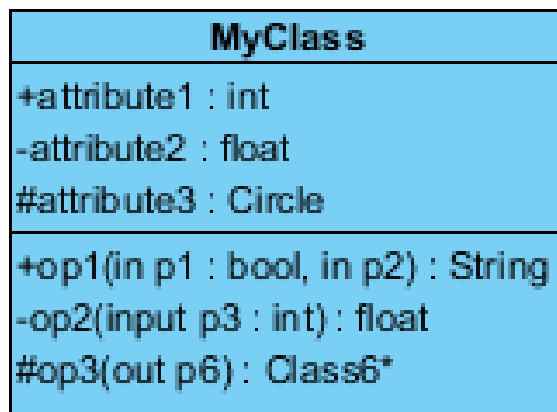
Класс белгілері үш бөлімнен тұрады:

- Класс атауы
- Класс атауы бірінші бөлімде пайда болады.

Класс атрибуттары:

- Атрибуттар екінші бөлімде көрсетілген.
- Атрибут түрі қос нүктеден кейін көрсетіледі.

- Атрибуттар кодтағы мүшелік айнымалыларға (мәліметтер мүшелеріне) сәйкес келеді.
- Класс операциялары (әдістері):
- Операциялар үшінші бөлімде көрсетілген. Бұл сынып ұсынатын қызметтер.
 - Әдістің қайтару түрі әдіс қолтаңбасының соңында қос нүктеден кейін көрсетіледі.
 - Әдіс параметрлерінің қайтару түрі параметр атауынан кейін қос нүктеден кейін көрсетіледі.
 - Класс әдісіндегі операциялардың картасы.



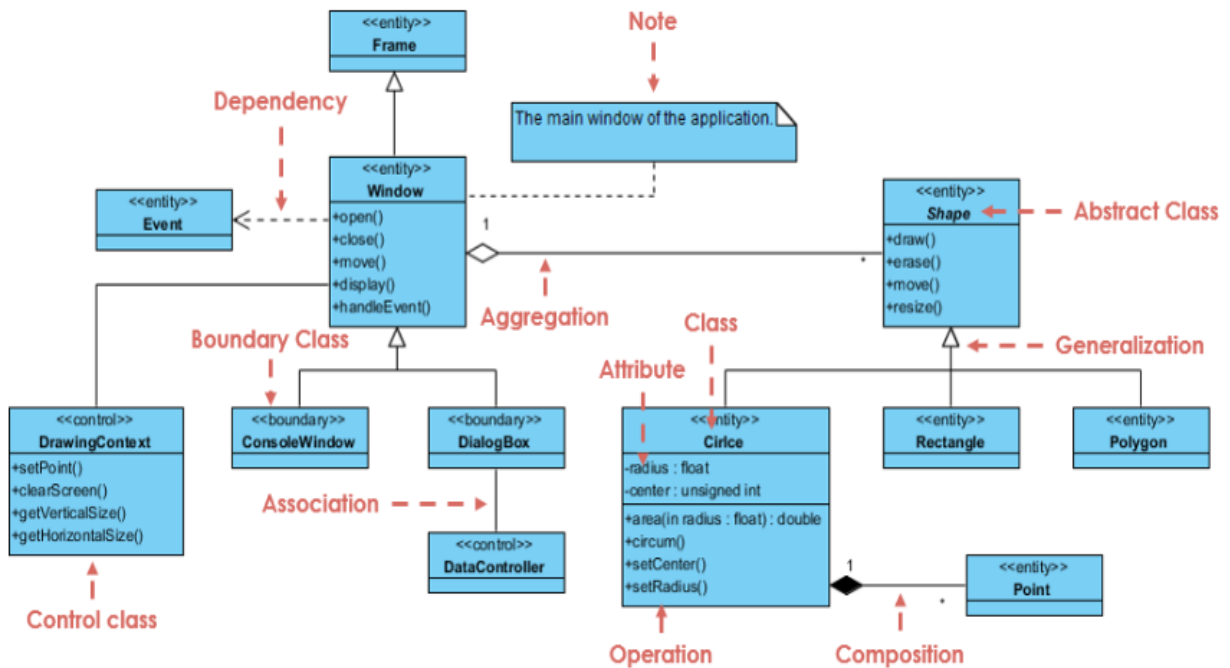
2.12-сурет – класс диаграммасының синтаксисі

Класстың графикалық көрінісі - MyClass жоғарыда көрсетілгендей:

- MyClass-те 3 атрибут және 3 операция бар;
- P3 op2 параметрі int типіне жатады;
- op2 өзгермелі мәнді қайтарады;
- op3 көрсеткіштің мәнін (a * арқылы белгіленген) Class6-ға қайтарады;

Класс басқа класстармен бір немесе бірнеше қатынастарға қатысуы мүмкін.

Класс диаграммасында класстарға немесе қатынастарға арналған жазбалар болуы мүмкін. Ескертпелер сұр түспен көрсетілген.



2.13-сурет – класс диаграммасының байланыстары

Жоғарыда көрсетілген диаграмманың мағынасын төмендегі тармақтар арқылы оқып түсіндіруге болады.

Пішін - бұл дерексіз сынып. Бұл курсивпен көрсетілген.

Пішін - бұл суперкласс. Шеңбер, төртбұрыш және көпбұрыш Shape-тен алынған. Басқаша айтқанда, шеңбер - бұл Shape. Бұл жалпылау / мұрагерлік қатынастар.

DialogBox пен DataController арасында байланыс бар.

Пішін - бұл терезенің бөлігі. Бұл агрегаттық қатынас. Пішін терезесіз болады.

Нүкте - шеңбердің бөлігі. Бұл композиция қатынасы. Шеңбер шеңберсіз мүмкін емес.

Терезе оқиғаға байланысты. Алайда, оқиға Терезеге тәуелді емес.

Шеңбердің атрибуттары - радиус және центр. Бұл субъект сыныбы.

Шеңбердің әдіс атаулары - аудан (), шеңбер (), `setCenter ()` және `setRadius()`.

Шеңбердегі радиус параметрі float типіндегі параметр болып табылады.

Сынып шеңберінің әдісі () double типінің мәнін береді.

Төртбұрыштың атрибуттары мен әдіс атаулары жасырылған. Диаграммадағы кейбір басқа сыныптарда олардың атрибуттары мен әдіс атаулары жасырылған.

Біз класс диаграммаларын бағдарламалық жасақтаманы әзірлеудің өмірлік циклінің әртүрлі кезеңдерінде қолдана аламыз және әдетте алға жылжу кезінде сынып диаграммаларын үш түрлі перспективада (нақтылау деңгейлері) біртіндеп модельдеу арқылы қолдана аламыз:

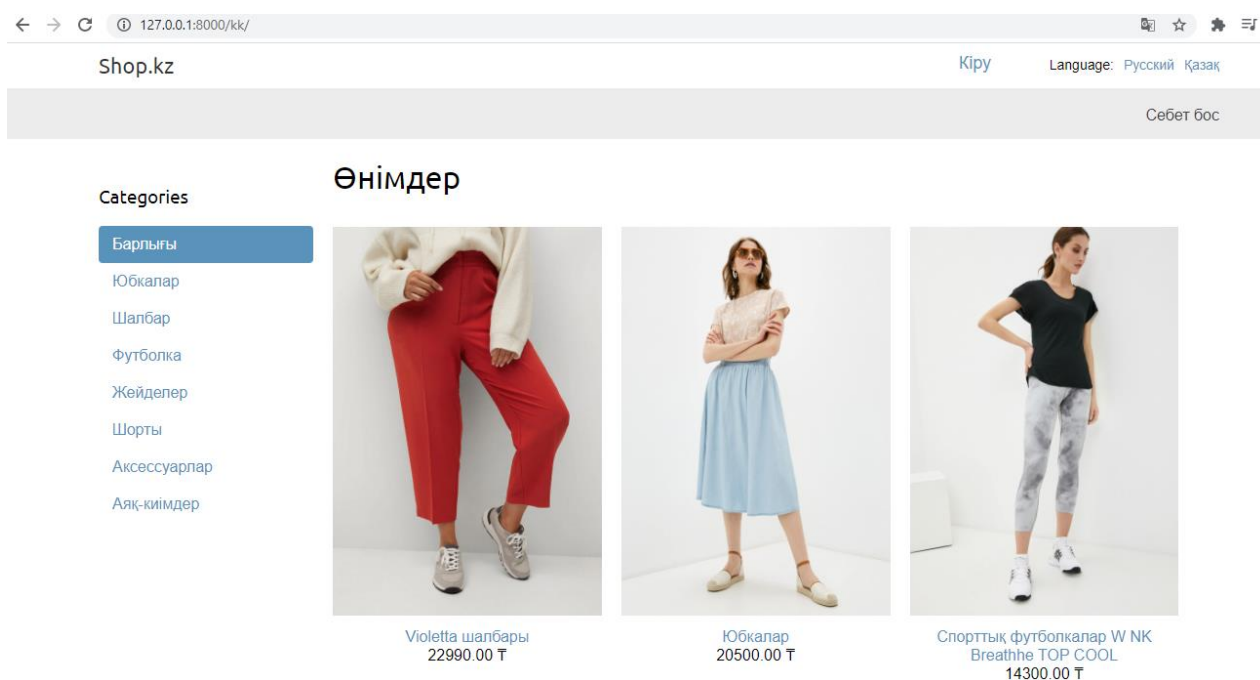
Тұжырымдамалық перспектива: Диаграммалар нақты әлемдегі заттарды сипаттау ретінде түсіндіріледі. Осылайша, егер сіз тұжырымдамалық перспективаны қарастыратын болсаңыз, онда зерттелетін саладағы тұжырымдамаларды көрсететін диаграмма жасаңыз. Бұл ұғымдар табиғи түрде оларды жүзеге асыратын сыныптарға қатысты болады. Концептуалды перспектива тілден тәуелсіз болып саналады.

Ерекшеліктің перспективасы: Диаграммалар бағдарламалық жасақтаманың абстракцияларын немесе сипаттамалары мен интерфейсі бар компоненттерін сипаттайды, бірақ нақты іске асыруды талап етпейді. Осылайша, егер сіз техникалық сипаттаманың перспективасын қарастыратын болсаңыз, онда біз бағдарламалық жасақтаманың интерфейстерін қарастырамыз, іске асыруды емес.

Іске асыру перспективасы: Диаграммалар бағдарламалық қамтамасыз етуді белгілі бір технология мен тілде сипаттау ретінде түсіндіріледі. Осылайша, егер сіз іске асырудың перспективасын қарастыратын болсаңыз, біз бағдарламалық қамтамасыз етуді енгізуді қарастырамыз.

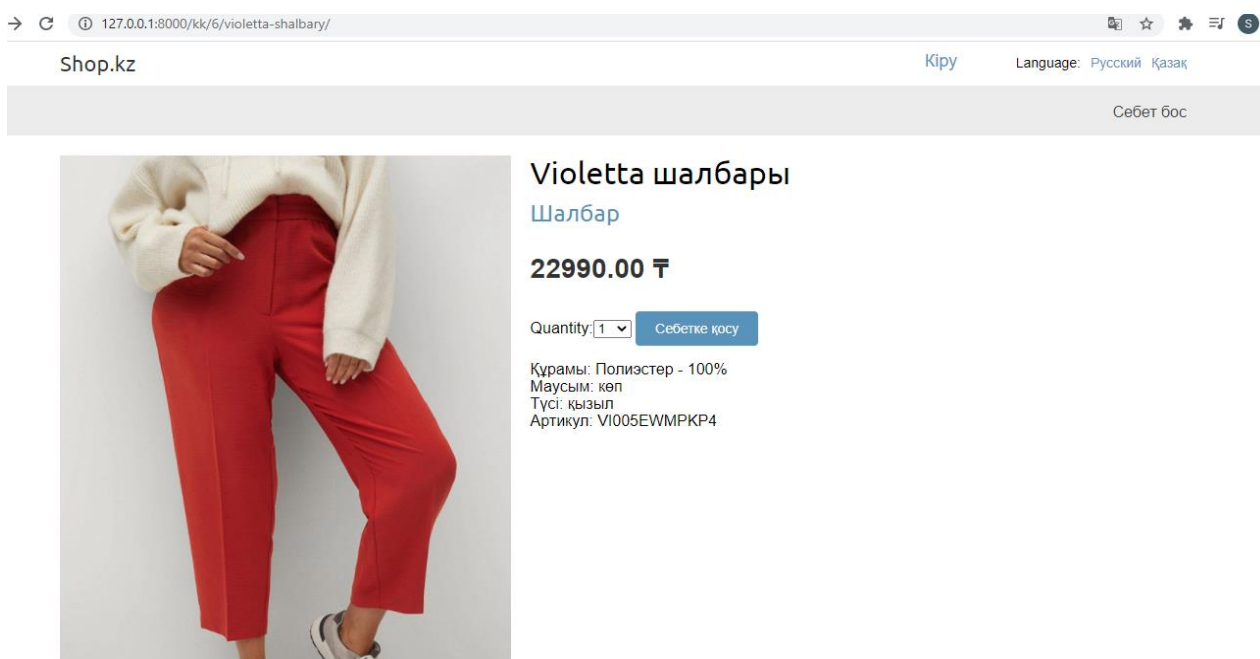
3 Жүзеге асыру және тестілеу бөлімі

3.1 Әйелдер киімдеріне арналған веб-сайт




3.1-сурет – Веб-сайттың басты беті

Веб-сайттың басты бетінде барлық киімдермен нав меню көрсетіледі. Ол жерде категорияға бөліп болады. Сонымен қатар, белгілі бір ұнаған киімді ашып толық ақпарат көруге болады.



3.2-сурет – Тауарды себетке қосу беті

Сіздің себетіңіз


Сурет	Өнім	Саны	Өшіру	Тауар өлшемінің бағасы	Бағасы
	Violetta шалбары	2	Update	Өшіру	₸22990.00
Барлығы					₸45980.00

Талонды қолдану:

Code: ҚолдануСауданы жалғастырыңыз

3.3-сурет – Себетке қосу процесі

Себетке қосқан сәтте біз таңдаған белгілі бір тауардың бағасы автоматты түрде санына қарай есептеліп көрсетіліп тұрады. Тауарды жеңілдікпен алу үшін арнайы талон нөмірін жазуға болады. Сол кезде біз тауарды жеңілдікпен аламыз.

	Violetta шалбары	2	Update	Өшіру	₸22990.00	₸45980.00
Таза бағасы					₸45980.00	
"777sss" (20%) купон					- ₸9196.00	
Барлығы					₸36784.00	

Талонды қолдану:

Code: ҚолдануСауданы жалғастырыңыз

3.4-сурет – Талонды қолдану сәті

Ары қарай тауарға тапсырыс беру үшін тапсырыс беру батырмасын басамыз. Тапсырыс беру сәті 3.5-суретте көрсетілген.

Тапсырыс беру

Аты:
Айдана

Фамилия:
Айдана

Электрондық пошта:
ss@mail.ru

Адрес:
АУЭС

Пошта индексі:
235515

Город:
Almaty

Беру

Сіздің тапсырысыңыз

- 2x Violetta шалбары
- "777sss" (20%) купон

₸45980.00
- ₸9196.00

Барлығы: ₸36784.00

3.5-сурет – Тапсырыс беру сәті

Тапсырыс берген соң тапсырысты нақтылау үшін қабылданды веб-беті ашылады.(3.6-сурет)

Shop.kz

Кіру

Language: Русский Қазақ

Себет бос

Рақмет

Сіздің тапсырысыңыз қабылданды. Сіздің тапсырыс идентификаторыңыз 16.

3.6-сурет – Тапсырыс қабылдау беті

Қабылданған тапсырысты көру үшін біз админ бетті қарауымыз керек. Админ беттің құрылымы толығырақ 3.7-суретте көрсетілген.

COUPONS

Coupons + Добавить ✎ Изменить

ORDERS

Orders + Добавить ✎ Изменить

SHOP

Categories + Добавить ✎ Изменить

Products + Добавить ✎ Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить ✎ Изменить

Пользователи + Добавить ✎ Изменить

Последние действия

Мои действия

- ✎ Nike Air MAX кроссовкасы
Product
- + Кроссовки Nike Air MAX
Product
- ✎ Maillen туфли
Product
- + Туфли Maillen
Product
- ✎ Аяқ-киімдер
Category
- + Обуви
Category
- ✎ Күн көзілдірігі - BOSCO
Product
- + Очки солнцезащитные - BOSCO
Product
- ✎ Liu Jo рюкзагы
Product
- + Рюкзак Liu Jo
Product

3.7-сурет – Админ беттің құрылымы

Бұл бетте біз деректер қорында құрылған моделдерді көре аламыз. Соның ішінде бізге керегі тапсырыстарды көру. Тапсырыстар бізде orders деген таблицанда сақталады.

Начало · Orders · Orders

Выберите order для изменения ДОБАВИТЬ ORDER +

Действие: ----- Выполнить Выбрано 0 объектов из 16

<input type="checkbox"/>	ID	FIRST NAME	LAST NAME	EMAIL	ADDRESS	POSTAL CODE	CITY	PAID	CREATED	UPDATED	ORDER DETAIL
<input type="checkbox"/>	16	Айдана	Айдана	ss@mail.ru	АУЭС	235515	Almaty	✘	23 мая 2021 г. 17:03	23 мая 2021 г. 17:03	View
<input type="checkbox"/>	15	Азилхан	Нуршайыков	ss@mail.ru	Almaty	23	2	✘	16 мая 2021 г. 8:30	16 мая 2021 г. 8:30	View
<input type="checkbox"/>	14	Азилхан	Нуршайыков	ss@mail.ru	Almaty	23	2	✔	16 мая 2021 г. 7:02	16 мая 2021 г. 7:02	View
<input type="checkbox"/>	13	Азилхан	Нуршайыков	ss@mail.ru	Almaty	23	2	✘	16 мая 2021 г. 7:00	16 мая 2021 г. 7:00	View
<input type="checkbox"/>	12	Азилхан	Нуршайыков	ss@mail.ru	Almaty	23	2	✘	16 мая 2021 г. 6:42	16 мая 2021 г. 6:42	View
<input type="checkbox"/>	11	Азилхан	Нвошайыков	ss@mail.ru	Almatv	23	2	✘	15 мая 2021 г. 15:14	15 мая 2021 г. 15:14	View

ФИЛЬТР

paid

- Все
- Да
- Нет

created

- Любая дата
- Сегодня
- Последние 7 дней
- Этот месяц

3.8-сурет – Orders админ беті

Жоғарыда көріп тұрғанымыздай, тапсырыстарды осындай үлгіде көре аламыз.

4 Экономикалық бөлім

4.1 Жобаның сипаттамасы

Бұл бөлімде «Әйелдер киімдеріне онлайн тапсырыс беруді ұйымдастыруға арналған веб-сайт әзірлеу» үшін әзірленген бағдарламалық өнімнің техникалық-экономикалық көрсеткіштері есептелген.

4.2 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу

Еңбек шығыны құрамдасын есептеудегі базалық көрсеткіш (4.1) формуламен есептелінеді:

$$Q = q \times c, \quad (4.1)$$

мұндағы, Q – шартты командалар саны;
 q – есеп түріне қарай шартты командалар санын ескеретін коэффициент;
 c – бағдарламаның қиындығы мен жаңалығын ескеретін коэффициент.

$$Q = q * c = 1700 * 1,30 = 2210$$

Атап өткен q коэффициентінің мәнін оқу құралындағы 1-кестеден таңдап алуға болады [1].

Атап өткен «с» коэффициенті оқу құралындағы 2-кестеден анықталады, ол күрделілік тобы бағанасы мен жаңалықтық дәрежесі бағанасының қиылысуы [1].

Бағдарламаның күрделілік коэффициенті $c=1,3$. Осы әдіспен базалық көрсеткішті табамыз: $Q = 1700$.

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек.

Жалпы бағдарлама жасауға кеткен уақыт әртүрлі компоненттен тұрады. Бағдарламалық өнімді жасауға кеткен жалпы уақыт құрамы оқу құралындағы 3-кестеде көрсетілген [1].

Уақыт адам-сағатпен есептеледі, ал T_d нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты Q командасының шартты санына байланысты есептік жолмен анықталады.

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтаймыз:

$T_{по}$ (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3-тен 5 күнге дейін, 8 сағаттан):

$$T_{по} = 24 \text{ адам / сағ.}$$

T_o (мақсат сипаттамасы уақыты) келесі (4.2) формуламен анықталады:

$$T_o = Q \times B / (50 \times K). \quad (4.2)$$

$$T_o = Q \times B / (50 \times K) = 2210 * 1,4 / (50 * 0,8) = 77,35 \text{ адам / сағ.}$$

мұндағы, B – мақсат есебі өзгерісінің коэффициенті, B коэффициенті мақсат күрделілігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін (2- кестені қарау);

K – бағдарлама жасаушы білектілігін ескеретін коэффициент (0,8) [1].

T_A (алгоритм құруға кеткен уақыт) (4.3) формуламен есептейміз:

$$T_A = Q / (50 \times K). \quad (4.3)$$

$$T_A = Q / (50 \times K) = 2210 / (50 * 0,8) = 55,25 \text{ адам / сағ.}$$

T_{bc} (блок – сұлба құруға кеткен уақыт) T_A сияқты (4.3) формуламен есептеледі.

T_n (бағдарламаның тілінде жазуға кеткен уақыт) келесі (4.4) формуламен анықталады:

$$T_n = Q \times 1,5 / (50 \times K). \quad (4.4)$$

$$T_n = Q \times 1,5 / (50 \times K) = 2210 * 1,5 / (50 * 0,8) = 82,9 \text{ адам / сағ.}$$

T_p (бағдарлама теру уақыты) келесі (4.5) формуламен анықталады:

$$T_p = Q / 50. \quad (4.5)$$

$$T_p = Q / 50 = 2210 / 50 = 44,2 \text{ адам / сағ.}$$

T_{ot} (бағдарламаны реттеу және тестілеу уақыты) келесі (4.6) формуламен анықталады:

$$T_{ot} = Q \times 4,2 / 50 \times K. \quad (4.6)$$

$$T_{ot} = Q \times 4,2 / 50 \times K = 2210 * 4,2 / 50 * 0,8 = 148,5 \text{ адам / сағ.}$$

T_d (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3-тен 5 күнге дейін, күніне 8 сағат):

$$T_d = 24 \text{ адам / сағ.}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде (4.7) формуламен есептеледі:

$$T = T_{\text{по}} + T_{\text{то}} + T_{\text{а}} + T_{\text{бс}} + T_{\text{н}} + T_{\text{п}} + T_{\text{от}} + T_{\text{д}}. \quad (4.7)$$

$$T = T_{\text{по}} + T_{\text{то}} + T_{\text{а}} + T_{\text{бс}} + T_{\text{н}} + T_{\text{п}} + T_{\text{от}} + T_{\text{д}} = 24 + 77,35 + 55,25 + 55,25 + 82,9 + 44,2 + 148,5 + 24 = 511,45 \text{ адам / сағ.}$$

4.3 Бағдарламалық қамсыздандыру шығынының есебі

Бағдарламалық қамсыздандыру шығыны ішіне еңбек ақы шығыны да, еңбек ақидан аударылымдар, амортизациялық және тағы да басқа шығындар кіреді, олар (4.8) формуламен анықталады:

$$C = \Phi OT + O_{\text{сн}} + A + C_{\text{ээ}} + C_{\text{мжк}} + C_{\text{то}} + C_{\text{пр}} + C_{\text{н}}, \quad (4.8)$$

Еңбек ақы екі жасаушыдан құрылады: негізгі еңбек ақы және қосымша еңбек ақы сомасы (немесе еңбек ақы қоры, ЕАҚ) негізгі еңбек ақы және қосымша еңбек ақы сомасы (4.9) формуламен есептеледі:

$$\Phi OT = Z_{\text{осн}} + Z_{\text{доп}}, \quad (4.9)$$

Негізгі еңбек ақы төмендегідей (4.10) формуламен анықталады:

$$Z_{\text{осн}} = T \times TC / (t_{\text{орг}} \times 8), \quad (4.10)$$

$$Z_{\text{осн}} = T \times TC / (t_{\text{орг}} \times 8) = 511,45 * 42500 / (21 * 8) = 129385 \text{ тг}$$

мұндағы, T – еңбек шығынының сомасы, (4.7) формуламен анықталады;
 $t_{\text{орг}}$ – бір айдағы орташа жұмыс күндерінің саны (21), жұмыс ұзақтығына көбейтіледі (8 сағат);

TC – тарифтік мөлшереме.

Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы (МӨЕА), 01.01.2019ж. бастап ҚР-да 42500 тенгені құрады.

Қосымша еңбек ақы негізгі еңбек ақының 20 % құрайды және келесі (4.11) формуламен есептеледі;

$$Z_{\text{доп}} = 0,2 \times Z_{\text{осн}}. \quad (4.11)$$

$$Z_{\text{доп}} = 0,2 \times Z_{\text{осн}} = 0,2 * 129385 = 25877 \text{ тг}$$

Жалпы еңбек ақысы (еңбек төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады:

$$\Phi OT = Z_{\text{осн}} + Z_{\text{доп}} = 129385 + 25877 = 155262 \text{ тг}$$

Әлеуметтік салық ЕАҚ 9,5 % құрайды (ҚР СК 358 б. 1-тарау) жұмыскердің табысынан, осы (4.12) формуламен есептеледі:

$$O_{\text{сн}} = (\text{ФОТ} - \text{ЗА}) \times 9,5\%, \quad (4.12)$$

$$O_{\text{сн}} = (\text{ФОТ} - \text{ЗА}) \times 9,5\% = (155262 - 15526,2) * 0,095 = 13275 \text{ тг}$$

мұндағы, ЗА – зейнетақы аударылымдар, ЕАҚ-нан 10% құрайды және әлеуметтік салықпен міндеттелмейді (4.13).

$$\text{ЗА} = \text{ЕАҚ} - 10\%. \quad (4.13)$$

$$\text{ЗА} = \text{ЕАҚ} - 10\% = 155262 * 0,1 = 15526,2 \text{ тг}$$

Амортизациялық аударылымдар амортизацияның тағайынды шамаларымен орындалады, пайыздармен жабдықтың баланстық құнына және (4.14) формуламен есептеледі:

$$A = B_{\text{бас}} A_{\text{ш}} \times N \times 100 \times 12 \times t, \quad (4.14)$$

мұндағы, $A_{\text{ш}}$ – амортизация шамалары;

$B_{\text{бас}}$ – жабдықтың бастапқы бағасы;

N – жұмыс орындалуына кеткен күннің саны;

t – дербес компьютерді қолдануға кеткен жалпы уақыт.

Амортизация шамалары ($A_{\text{ш}}$), (4.15) формуламен есептеледі:

$$A_{\text{ш}} = B_{\text{бас}} - K_{\text{тар}} T_{\text{норм}} * B_{\text{бас}} * 100\%, \quad (4.15)$$

$$A_{\text{ш}} = B_{\text{бас}} - K_{\text{тар}} T_{\text{норм}} * B_{\text{бас}} * 100\% = 374980 - 187494 * 374980 * 100\% = 23,75\%$$

мұндағы, $K_{\text{тар}}$ – таратылым құны, жабдықтың құнынан 5% құрайды.

$$K_{\text{тар}} = 374980 * 0,05 = 18749 \text{ тг.}$$

$$\text{Құны } B_{\text{бас}} = 369990 + 4990 = 374980 \text{ тг.}$$

Компьютердің қазіргі уақыттағы құны 369990 тг [3].

Тінтуір (мышка) құны 4990 тг [3].

$T_{\text{норм}}$ – жабдықтың нормативтік қызмет ету мерзімі (дербес компьютер үшін – 4 жыл).

$$N = T / 24 = 511,45 / 24 = 21,3 \text{ күн.}$$

Бұл жерде T еңбек шығынына тең, оның мәні – 511,45 адам/сағ тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты (4.16) формуламен есептеледі:

$$T = T_A + T_{bc} + T_H + T_{II} + T_{OT} \quad (4.16)$$

$$T = T_A + T_{bc} + T_H + T_{II} + T_{OT} = 77,35 + 55,25 + 55,25 + 82,9 + 44,2 + 148,5 = 463,45 \text{ адам / сағ}$$

Амортизациялық аударылымдар:

$$A = 374980 * 23,75 * 21,3100 * 12 * 463,45 = 341 \text{ тг}$$

Электрэнергия шығындары (4.17) формуламен есептеледі:

$$C_{ЭЭ} = Q \times k_3 \times T \times C_{кВт-сағ}, \quad (4.17)$$

$$C_{ЭЭ} = Q \times k_3 \times T \times C_{кВт-сағ} = 0,45 * 0,8 * 12,5 * 463,45 = 2085 \text{ тг}$$

мұндағы, Q – ЭЕМ қуаты (450 Вт = 0,45 кВт);

k_3 – жүктеме коэффициенті (0,8);

$C_{кВт-сағ}$ – 1 кВт-сағ электрэнергиясының құны;

T – жұмыс уақыты, сағ.

1 кВт-сағ электрэнергиясының құны 12,5 тг.

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ($C_{МжК}$), сонымен қатар техникалық қызмет көрсету шығыны ($C_{ТО}$), жабдықтың құнынан 1,5% және 2,5% құрайды және формулалар мен есептеледі (4.18-4.19):

$$C_{МжК} = 0,015 \times C_{обор}, \quad (4.18)$$

$$C_{МжК} = 0,015 \times C_{обор} = 0,015 \times 374980 = 5625 \text{ тг}$$

$$C_{ТО} = 0,025 \times C_{обор}, \quad (4.19)$$

$$C_{ТО} = 0,025 \times C_{обор} = 0,025 \times 374980 = 9375 \text{ тг}$$

Басқару мен қызмет көрсетуге байланысты үстеме шығындар, сондай-ақ жабдықты пайдалану кезіндегі және де кәсіпорын үдерістері мен айналымдарынан қосымша шығындар еңбек ақы қорынан 50% құрайды және де (4.20) формуламен есептеледі [1]:

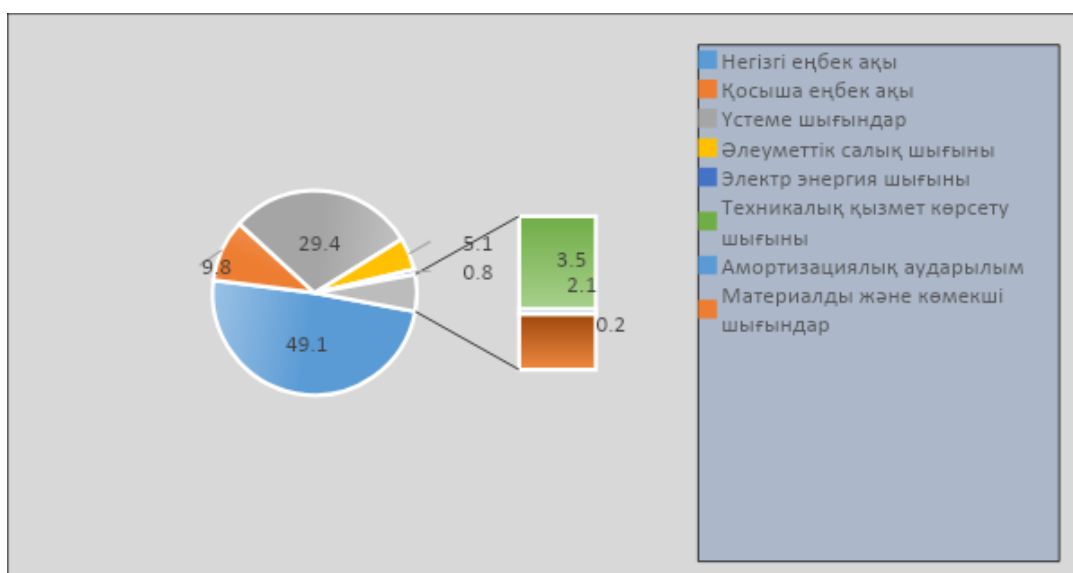
$$C_H = 0,5 \times \text{ЕАҚ}, \quad (4.20)$$

$$C_H = 0,5 \times \text{ЕАҚ} = 0,5 \times 155262 = 77631$$

Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелерін кесте түрінде ұсыну керек (4.1-кестені қараңыз), шығын статьясын атап, және оның ортақ құндағы сыбағаларын пайызбен есептеп, бағдарламалық қамсыздандырудың өзіндік құнының құрылымының диаграммасын сызу керек.

4.1-кесте – Өзіндік құнның қорытынды кестесі

Шығын бабы атауы		Сомасы, теңге	Әр баптың үлесі, %
ЕАҚ	ЕА _{нег}	129385	49,1
	ЕА _{қос}	25877	9,8
Үстеме шығындар, С _{Накл}		77631	29,4
Әлеуметтік салық шығыны, ӘС		13275	5,1
Пайдалану шығындары	С _{ээ}	2085	0,8
	С _{ТО}	9375	3,5
	А _{жыл}	341	0,2
Материалдар және көмекші, С _{Мжк}		5625	2,1
Барлығы:		263594	



4.1-сурет – Өнімнің өзіндік құнының диаграммасы

Бағдарламалық жасақтама өнімінің бағасы – шығындар мен пайдадан анықталады, ол өз кезегінде өзіндік құнның 30 пайызын (шартты түрде) құрайды (4.21 формула бойынша).

$$BAFA = C + (C * 30/100), \quad (4.21)$$

$$BAFA = C + (C * 30/100) = 263594 + (263594 * 0.3) = 342672,2 \text{ тг.}$$

Қазақстан Республикасында қосылған құн салығының (ҚҚС) ставкасы 12% құрайды (4.22 формула бойынша).

$$C_c = \text{БАҒА} + \text{БАҒА} * 12\%, \quad (4.22)$$

$$C_c = 342672,2 + 342672,2 * 0,12 = 383793 \text{ тг.}$$

4.4 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі

Бағдарлама өнімін сатып алуға және оны өндіріске енгізу шығындары келесі шығындардан тұрады (4.23):

$$\Sigma Z = C_c + C_{\text{тр}} + C_o, \quad (4.23)$$

Бағдарламаның бағасы: $C_c = 383793$ тг

мұндағы, C_c – жүйенің құны, мың тенге;

$C_{\text{тр}}$ – көлік шығыны, жүйе құнынан – 25 %, мың тенге;

C_o – өнімді игеруге деген шығыннан, мың тенге.

$$C_{\text{тр}} = 0,25 * C_c = 383793 * 0,25 = 95948 \text{ тг.}$$

Жүйе құны үстінде есептелінді, ал қалғандары келесі түрде есептеледі. Өнімді игеруге деген маманды оқыту шығыны, оқытуға кеткен уақыт пен оған деген консалтингті фирмадағы мөлшерлемеден тұрады (4.24):

$$C_o = T * C_{\text{оп}}, \quad (4.24)$$

$$C_o = T * C_{\text{оп}} = 4 * 8 * 2600 = 83200$$

мұндағы, T – оқытуға кеткен уақыт, сағат.

$C_{\text{оп}} = 2600$. Қазіргі таңда сағатына 2500-3000 тг аралығында.

$$\Sigma Z = C_c + C_{\text{тр}} + C_o = 383793 + 95948 + 83200 = 562941 \text{ тг.}$$

Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебін 4.2-кестеге келтіру керек.

4.2-кесте – Ақпарат жүйелерін енгізуге керекті бір жолымғы шығындар есебінің жиынтығы

Шығын бабы атауы	Сомасы, тенге
Жүйенің құны	383793
Көлік шығыны	95948
Жүйені оқуға кеткен шығыны	83200
Барлығы:	562941

4.5 Игеру саласындағы жылдық бір жолғы шығындар есебі

Ақпараттық технологиялар (АТ) енгізуде пайдалану шығындарының өзгеруіне өте үлкен мән қойылады. Фирма шығындары жүйенің зерттеме және

тираждауына шығындары кіреді. Сондықтан пайдалану шығындарын АТ енгізу алдында және кейін де есептеген жөн [2].

Ақпараттық технологияларын қолданған кездегі жылдық шығындары келесі баптардан тұрады:

- жылдық ЕАҚ;
- әлеуметтік салық аударым;
- басқа да шығындар.

Осының барлығысын енді (4.25) формула арқылы жазайық:

$$C_{\Sigma} = 3П + O_{\text{сн}} + П_{\text{р}}, \quad (4.25)$$

мұндағы, 3П – жылдық еңбек ақы шығыны, мың теңге;

$O_{\text{сн}}$ – әлеуметтік салық аударым, мың теңге;

$П_{\text{р}}$ – басқа да шығындар, мың теңге.

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі (4.26) формуламен есептеледі:

$$3П = (O_c \times t \times K_p) \times Ч \times (1 + K_d), \quad (4.26)$$

мұндағы, O_c – маманның сағаттық ақысы, 3000 мың теңге/сағ.;

t – жұмыс күнінің ұзақтығы, 8 сағ.;

K_p – жылдағы жұмыс күндер саны, 245 күн;

$Ч$ – үдеріске қатысы бар мамандар саны, адам;

K_d – қосымша еңбек ақы коэффициенті, 20% [1].

$$3П = (O_c \times t \times K_p) \times Ч \times (1 + K_d)$$

$$3П = (3000 * 8 * 245) * 2 * (1 + 0,2) = 14112000 \text{ тг.}$$

$$3А = \text{ЕАҚ} * 10\% = 14112000 * 0,1 = 1411200 \text{ тг.}$$

$$O_{\text{сн}} = (\text{ФОТ} - 3А) * 9,5\% = (14112000 - 1411200) * 0,095 = 1206576 \text{ тг.}$$

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі (4.27) формуламен есептеледі:

$$П_{\text{р}} = 3П * 0,3, \quad (4.27)$$

$$П_{\text{р}} = 3П * 0,3 = 1206576 * 0,3 = 361973 \text{ тг.}$$

$$C_{\Sigma} = 3П + O_{\text{сн}} + П_{\text{р}} = 14112000 + 1206576 + 361973 = 15680549 \text{ тг.}$$

Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындарды келесі кестеге сомасын және әр баптарын көрсетіп толтырыңыз (4.3-кестені қараңыз).

4.3-кесте – Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, теңге
Жылдық ЕАҚ	14112000
Әлеуметтік салық аударым	1206576
Басқа да шығындар	361973
Барлығы:	15680549

4.6 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі

Еңбек өнімділігі өсуінен түскен үнемділікті келтірінді шығындардың базалық (C_0) және ұсынылған (C_1) нұсқалар айырмасы ретінде шығарып алуға болады (4.28).

$$\Delta_{\text{yr}} = C_0 - C_1, \quad (4.28)$$

мұндағы, C_0 – база мезгілінде жылдық келтірінді шығындар (қол жұмысын қолданған кезде), мың теңге;

C_1 – ұсынылған мезгілінде жылдық келтірінді шығындар (бағдарламалық өнімді енгізгеннен кейін), мың теңге.

Ұйымда жүйе енгізілмей тұрғанда (қол еңбегімен айналысқанда) жылдық келтірінді шығындар келесі баптардан тұрады:

- еңбек ақы қоры;
- әлеуметтік салық;
- басқада шығындар.

Ұсынылған мезгілінде жылдық келтірінді шығындарды (C_1) есептеу:

Осының барлығысын енді (4.25-4.27) формула арқылы жазайық.

$$C_{\Sigma} = 3\Pi + O_{\text{CH}} + \Pi_p, \quad (4.29)$$

$$C_{\Sigma} = 3\Pi + O_{\text{CH}} + \Pi_p = 14112000 + 1206576 + 361973 = 15680549 \text{ тг}$$

База мезгілінде жылдық келтірінді шығындарды (C_0) есептеу:

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі (4.30) формуламен есептеледі:

$$3П = (O_C \times t \times K_P) \times Ч \times (1 + K_D), \quad (4.30)$$

$$3П = (3000 * 8 * 245) * 3 * (1 + 0,2) = 21168000$$

$$3А = ЕАҚ * 10\% = 21168000 * 0,1 = 2116800 \text{ тг}$$

$$ОСН = (ФОТ - 3А) \square 9,5\% = (21168000 - 2116800) * 0,095 = 1809864 \text{ тг}$$

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі (4.31) формуламен есептеледі:

$$П_P = 3П \times 0,3, \quad (4.31)$$

$$П_P = 3П \times 0,3 = 21168000 * 0,3 = 6350400 \text{ тг}$$

$$С_э = 3П + О_{сн} + П_P = 21168000 + 1809864 + 6350400 = 29328264 \text{ тг}$$

База мезгілінде жылдық келтірінді шығындар $C_0 = 29328264$ тг

Ұсынылған мезгілінде жылдық келтірінді шығындар $C_1 = 15680549$ тг

$$\mathcal{E}_y = C_0 - C_1 = 29328264 - 15680549 = 13647715 \text{ тг}$$

Базалық мерзімде жылдық бір жолдық шығындарды келесі кестеге сомасын, баптарын көрсетіп толтырыңыз (4.4-кестені қараңыз).

4.4-кесте – Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, теңге
Жылдық ЕАҚ	21168000
Әлеуметтік салық аударым	1809864
Басқа да шығындар	6350400
Барлығы:	29328264

Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі, яғни $\mathcal{E}_y = 13647715$ тенгені құрайды.

4.7 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу

Нормативтік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті келесі (4.32) формуламен есептеледі:

$$E_H = 1 T_H, \quad (4.32)$$

$$E_H = 13 = 0,33$$

мұндағы, T_n – нормативтік күрделі қаржы салымын өтелу мерзімі, жыл.

Нормативтік күрделі қаржы салымын өтелу мерзімі, АТ моральдық тозуы техникалық құралдардың және жоба шешімдерінің тозуына байланысты ($T_n = 1, 2, 3 \dots n$) бағдарлама өнімдерінің өтеу мерзімі 3 жыл.

Есептік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті (4.33):

$$E_p = \frac{\mathcal{E}_{y\%}}{K}, \quad (4.33)$$

мұндағы, K – ұсынылған мезгілінде жылдық келтірінді шығындар $C1 = K = 15680549$ тенге.

$$E_p = \frac{\mathcal{E}_{y\%}}{K} = 1364771515680549 = 0,87$$

Есептік күрделі қаржы салымын өтелу мерзімі (4.34):

$$T_p = \frac{1}{E_p}. \quad (4.34)$$

$$T_p = \frac{1}{E_p} = 10,87 = 1,15$$

Дисконтты есептемегенде өтелу мерзімі 1,15 жыл.

Ақпараттық жүйелер енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштерін есептеу қорытындыларын келесі кестеге сомасын, баптарын көрсетіп толтырыңыз (4.5-кестені қараңыз).

4.5-кесте – Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері

Көрсеткіштер атауы	Мәні
Шартты жылдық шығынды үнемдеу, мың тенге	13647715
Күрделі қаржы салымының экономикалық тиімділігінің коэффициенті (E_p)	1,15
Күрделі қаржы салымын өтелу мерзімі (T_p), жыл	0,87

4.8 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау

Таза дисконттық табыс (ЧДД) келесі (4.35) формуламен есептеледі:

$$\text{ЧДД} = \sum_{t=0}^T (P_t - Z_t) \times \frac{1}{(1+E)^t} - K, \quad (4.35)$$

$$\mathcal{E}_t = (P_t - Z_t) = \mathcal{E}_{yr} = 13647715 \text{ теңге.}$$

мұндағы, P_t – ұсынылған ақпараттық жүйелерді енгізудегі болжамды нәтиже, теңге;

Z_t – ақпараттық жүйелерді жасауға және пайдалануға керекті болжамды шығындар, теңге;

$\mathcal{E}_t = (P_t - Z_t)$ – t -аралықты есептеуде жеткен әсер;

K – күрделі қаржы салымы;

t – есептеу аралығының нөмірі ($t = 0, 1, 2, \dots, T$);

T – есеп шегі;

E – тұрақты дисконт мөлшері, 20%.

Есеп аралығы – есептеу кезеңіндегі уақыттың бір бөлігі (жыл, квартал, ай және т. б.).

Егер инвестициялық жобадан таза дисконттық табыс (ЧДД) оң болса, жоба тиімді (берілген дисконттау мөлшерінде).

Онда барлық есеп шегін ескергендегі таза дисконттық табыс сомасы:

$$\text{ЧДД} = \mathcal{E}_1 \frac{1}{1+E} + \mathcal{E}_2 \frac{1}{(1+E)^2} + \mathcal{E}_3 \frac{1}{(1+E)^3} - K ;$$

$$\begin{aligned} \text{ЧДД} &= \mathcal{E}_1 \frac{1}{1+E} + \mathcal{E}_2 \frac{1}{(1+E)^2} + \mathcal{E}_3 \frac{1}{(1+E)^3} - K \\ &= 136477151,2 + 136477151,22 + 136477151,23 - 15680549 = 11373096 + 9477580 + 7897983 - 15680549 = 13068110 \\ &\text{тенге} \end{aligned}$$

Таза дисконттық табыстың оң мәні $\text{ЧДД} > 0$, инвестицияның тиімділігін көрсетеді, ақпараттық жүйелер керекті көлемде пайда алып келеді.

Табыстық индексі (ИД) келтірілген әсерлердің сомасының күрделі қаржы салымына қатынасы. Ол келесі (4.36) формуламен есептеледі:

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1+E)^t}, \quad (4.36)$$

мұндағы, K – күрделі қаржы салымы немесе инвестицияның құны.

$$\begin{aligned} \text{ИД} &= 136477151 + 0,2 + 136477151 + 0,22 + 136477151 + 0,23 \\ &15680549 = 11373096 + 9477580 + 7897983 - 15680549 = 28748659 - 15680549 = 1,83 \end{aligned}$$

Инвестиция тиімді деп есептеледі, егер табыстық индексі бірден көп болса, $\text{ИД} > 1$, онда ақпараттық жүйелерге деген инвестиция тиімді.

Егер $\text{ИД} > 1$, жоба тиімді, егер $\text{ИД} < 1$ – тиімді емес

Ішкі табыстық мөлшері (ВНД):

Ішкі табыстық мөлшері (ВНД немесе IRR) – келтірілген әсер, келтірілген күрделі қаржы салымдарға тең болғандағы дисконттау мөлшерін ($r_{\text{вн}}$) көрсетеді. Бұл көрсеткіш $r_{\text{вн}}$ (ВНД) келесі теңдеуден шығады (4.37):

$$\sum_{t=0}^T \frac{P_t - Z_t}{(1 + r_{\text{вн}})^t} = \sum_{t=0}^T \frac{K_t}{(1 + r_{\text{вн}})^t} \quad (4.37)$$

Егер $r_{\text{вн}}$ мәні инвесторға керек табыс мөлшері салынған капиталдан көп не ішкі табыстық мөлшері (ВНД).

егер $E_1 \rightarrow \text{ЧДД}_1 > 0$;
 $E_2 \rightarrow \text{ЧДД}_2 < 0$;

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 - \text{ЧДД}_2} \cdot (E_2 - E_1);$$

егер $E_1 \rightarrow \text{ЧДД}_1 > 0$;
 $E_2 \rightarrow \text{ЧДД}_2 > 0$;

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 + \text{ЧДД}_2} \cdot (E_2 - E_1);$$

$E_1 = 0,20$ тең болған кездегі ЧДД-ның есептелуі:

$$\text{ЧДД}_1 = 136477151 + 0,20 + 13647715(1 + 0,20)^2 + 13647715(1 + 0,20)^3 - 15680549 = 11373096 + 9477580 + 7897983 - 15680549 = 13068110 \text{ тенге}$$

$E_2 = 0,25$ тең болған кездегі ЧДД-ның есептелуі:

$$\text{ЧДД}_2 = 136477151 + 0,25 + 13647715(1 + 0,25)^2 + 13647715(1 + 0,25)^3 - 15680549 = 10918172 + 8734538 + 6987630 - 15680549 = 10959791 \text{ тенге}$$

$E_1 \rightarrow \text{ЧДД}_1 > 0$ және $E_2 \rightarrow \text{ЧДД}_2 > 0$ тең болғандықтан мына формула бойынша есептейміз:

$$\text{IRR} = 0,20 + \frac{13068110}{13068110 + 10959791} \cdot (0,25 - 0,20) = 0,227$$

Сонымен, дисконттау мөлшері (20 % ... 25 %) осы аралыққа тиесілі.

4.6-кесте – Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері

Көрсеткіштер атауы	Мәні
Бағдарлама өнімін әзірлеуге және енгізуге шығын, мың теңге	15680549

4.6- кестенің жалғасы

Бағдарлама өнімін енгізгеннен кейінгі болжалды үнем, мың теңге	13647715
Таза дисконттық табыс, мың теңге	13068110
Табыстық индекс	1,83
Ішкі табыстық мөлшері	0,227 = 22,7%
Дисконтталған өтелу мерзімі, жыл	1,15
Моральдық ескіру мерзімі, жыл	3

Есептеп шығарылған көрсеткіштер бағдарлама өнімін енгізудің экономикалық тиімді екенін көрсетеді, тест қорытындыларын өңдеуге кететін уақыттың азаюы жылдық шығындар 15680549 теңгеге азайатынын дәлелдейді. Сондықтан бағдарлама өнімін әзірлеу және енгізу экономикалық пайдалы.

4.9 Экономикалық бөлім бойынша қорытынды

Техникалық-экономикалық негіздемеде шығындар сметасы жасалып, осы бағдарламалық құралды әзірлеуге байланысты шығыстар сомасының есебі орындалды.

Құқық қорғау қызметінде биометриялық технологияларды қолданудың нақты экономикалық нәтижесін болжау өте қиын. Алайда, құқық қорғау саласында биометриялық технологияларды қолдану кезінде алынатын ақпараттық артықшылық заңсыз көші-қон, жеке басын куәландыратын құжаттарды қолдан жасау және т.б. сияқты бірқатар қылмыстық әрекеттердің алдын алады.

Компаниялар тәуекелдерді азайту, биометриялық жүйені тиімді және ыңғайлы ету үшін осы технологияны пайдалану жағдайларын белсенді түрде қарастыруы керек, сонымен қатар қолданыстағы жүйелердің қауіпсіздігін жақсарту туралы ғана емес, сонымен қатар жақын болашақта бәсекелестік артықшылықты қалай қамтамасыз ету туралы ойлануы керек.

5 Өміртіршілік қауіпсіздігі

«Өмір-тіршілік қауіпсіздігі» – адам өміріне төнетін жалпы қауіп-қатерлерді зерттейтін, соған қарсы қоюға болатын шараларды әзірлейтін ғылыми білімнің саласы. Ол нақты машиналардың, құрал-жабдықтардың, механизмдердің, өндірістің қауіпсіздігін қамтамасыз ету мәселелерін шешпейді, онымен арнайы мамандықтар айналысады (атом қауіпсіздігі, электр қауіпсіздігі, жаралыс, өрт қауіпсіздігі және т.с.с). «Өмір-тіршілік қауіпсіздігі» пәнінің мақсаты – қауіпсіздік саласында жалпы біліктілікті көтеру, адамдардың ойлау қабілетін өмірдің қауіпсіздігін қамтамасыз етуге бағыттау -осылар барлық арнайы мамандықтардың қауіпсіздігінің ғылыми-әдістемелік іргесі болып саналады. Осы курсты оқып өткен адамның өз ортасындағы қауіпті және өзі мен өзгеге зиянын тигізетін құбылыстарды тануға, соған қарсы білікті әрекет жасауына мүмкіншілігі болады.

Дипломдық жобаның негізгі мақсаты Алматы облысының Сарқанд кентінде GPON технологиясын қолдану арқылы мультисервистік желіні ұйымдастыру болып табылады. Бұл дипломдық жобаны жасау барысы компьютерлік технологиялар арқылы жүзеге асырылды. Жұмыс барысында адам өміріне қауіп тудыратын факторлардың алдын алу жағдайлары қарастырылды. Қарастырылатын жұмыс орны ретінде осы жобаны жасау барысында қызметкерлер отырған бөлмені қарастырдым.

5.1 Жұмыс орнының эргономикасы және еңбек жағдайларын талдау

Ғылыми-техникалық прогрестің дамуымен адамдардың өз еңбек міндеттерін қауіпсіз орындау мүмкіндігі маңызды рөл атқарады.

Осыған байланысты адамның еңбек қауіпсіздігі және өмір сүру қауіпсіздігі туралы ғылым құрылып, дамып келеді.

Өмір тіршілігінің қауіпсіздігі (БЖД) - бұл өмір сүру ортасындағы адамның қауіпсіздігін қамтамасыз етуге, оның денсаулығын сақтауға, зиянды және қауіпті факторлардың рұқсат етілген мәндерге дейін әсерін төмендету жолымен қорғау әдістері мен құралдарын әзірлеуге, бейбіт және соғыс уақытындағы төтенше жағдайлардың салдарын жоюдағы залалды шектеу жөніндегі шараларды әзірлеуге бағытталған іс-шаралар кешені.

Еңбекшілердің денсаулығын қорғау, еңбек жағдайларының қауіпсіздігін қамтамасыз ету, кәсіптік аурулар мен өндірістік жарақаттануды жою адам қоғамының басты қамқорлығының бірін құрайды. Еңбекті ғылыми ұйымдастырудың прогрессивті нысандарын кеңінен қолдану қажеттілігіне, қол, біліктілігі аз еңбекті барынша азайтуға, кәсіби аурулар мен өндірістік жарақаттануды болдырмайтын жағдайды жасауға назар аударылады.

Жұмыс орнында өндірістің қауіпті және зиянды факторларының ықтимал әсерінен қорғау шаралары көзделуі тиіс.

Бұл факторлардың деңгейлері құқықтық, техникалық және санитарлық-техникалық нормаларда айтылған шекті мәндерден аспауы тиіс.

Бұл нормативтік құжаттар жұмыс орнында жұмыс істеушілерге қауіпті және зиянды факторлардың әсері жойылған не мүлдем жойылған не рұқсат етілген шектерде болатын еңбек жағдайларын жасауға міндеттейді.

5.1.1 Монитормен жұмыс

Электрондық-сәулелі түтік (ЭЛТ) - бұл электрондық зеңбірек. Бұл дегеніміз, ЭЛТ теріс зарядталған, демек, ЭЛТ-дан тыс оң зарядталған бөлшектердің жиналуы.

Адам қоршаған ортада оң және теріс иондардың арақатынасы бірдей болғанда жақсы сезінеді. Алайда, монитордың экраны алдында оң иондар артық пайда болады. Әрқашанда офистің ауасындағы шағын бөлменің (шаң, темекі түтіні және т. б.), оң зарядталған иондардың ағынымен айдалады және экран алдында отырған программисттің беті мен көздерінде тұрады.

Мұндай "бомбалаудың" нәтижесінде программистте пайда болуы мүмкін:

- бас ауруы, ұйқысыздық;
- көздің шаршауы;
- терінің тітіркенуі.

5.1.2 Көрініс әсері және жұмыс үстелі

Ашық боялған кеңсе жиһазы және үлкен терезелер қосымша жарық көздері болып табылады. Өте жарық бөлмеде монитор экранындағы әріптер мен сандар нашар көрінеді. Бұл бас ауруы, көрудің нашарлауы, концентрацияның төмендеуі, сондай-ақ ақпараттың дұрыс қабылданбауынан жұмыста қателіктерге әкеледі.

5.1.3 Шу

Шуылдың жоғары деңгейі түсті сигналдарды тануда қиындықтар туғызады, түсті сигналдарды қабылдау жылдамдығын төмендетеді, түсті қабылдау жылдамдығын, көру өткірлігін, көру бейімделуін төмендетеді, көрнекі ақпаратты қабылдауды бұзады, үйлестірілген әрекеттерді тез және нақты орындау қабілетін төмендетеді, еңбек өнімділігін 5-10% азайтады. Дыбыс қысымы 90 Дб жоғары шу деңгейінің ұзақ әсері еңбек өнімділігін 30-60% төмендетеді. Программистерді медициналық тексеру еңбек өнімділігінің төмендеуінен басқа жергілікті әрекет кезінде шудың жоғары деңгейі шаршауға, есту қабілетінің нашарлауына әкеп соқтыратынын көрсетті. Сонымен қатар, жалпы әрекет кезінде шудың жоғары деңгейі жүрек қызметінің ырғағының бұзылуына, қан қысымының өзгеруіне, тыныс алу органдарының нашарлауына әкеледі. Бөлмедегі шу көздері баспа құрылғылары болып табылады.

5.1.4 Артық жылу бөлу

Сыртқы ортаның жоғары температурасы тез шаршауға алып келеді, көру және есту ақпаратын қабылдау жылдамдығын, жүрек қызметінің бұзылуы (жүрек соғуының жылдамдығының артуы), қан қысымының өзгеруі салдарынан адамның жалпы тежелуін төмендетеді.

Көптеген бағдарламашылар ақыл - ойдың шамадан тыс күшеюі, көру және есту аппараттарының шамадан тыс күшеюі, еңбектің монотондылығы, эмоциялық артық жүктеме сияқты психофизикалық факторлардың әсерімен байланысты. Көрсетілген қолайсыз факторлардың әсері дамып келе жатқан шаршаудан туындайтын жұмысқа қабілеттіліктің төмендеуіне әкеледі. Шаршаудың пайда болуы мен дамуы орталық жүйке жүйесінің жұмыс процесінде пайда болатын өзгерістермен, бас миының қабығындағы тежеу процестерімен байланысты.

5.1.5 Жұмыс орнының жарықтандырылуы

Жұмыс орнын ұтымды жарықтандыру адамның еңбек қызметінің тиімділігіне әсер ететін, жарақаттану мен кәсіби аурулардың алдын алатын маңызды факторлардың бірі болып табылады. Дұрыс ұйымдастырылған жарықтандыру қолайлы еңбек жағдайын жасайды, еңбек қабілеттілігі мен өнімділігін арттырады. Программистің жұмыс орнындағы жарықтандыру қызметкер өз жұмысын кернеусіз орындай алатындай болуы тиіс. Көру мүшелерінің шаршауы бірқатар себептерге байланысты:

- 1) жарықтың жеткіліксіздігі;
- 2) шамадан тыс жарықтандыру;
- 3) жарықтың дұрыс емес бағыты.

Жарықтандырудың жеткіліксіздігі көру кернеуіне алып келеді, назар аударуды әлсіретеді, уақытынан бұрын шаршаудың туындауына алып келеді. Жарықтың шамадан тыс ашық болуы көзді соққылау, тітіркену және кесу тудырады.

Жұмыс орнында жарықтың дұрыс емес бағыты күрт көлеңкелер, құймақ жасай алады, жұмыс істеушіге бағдар бере алады. Осы себептердің барлығы жазатайым оқиғаға немесе кәсіби ауруға әкелуі мүмкін, сондықтан жарықтың дұрыс есебі маңызды.

5.1.6 Микроклимат параметрлері

Микроклимат параметрлері кең шектерде өзгеруі мүмкін, ал адамның тіршілік әрекетінің қажетті шарты терморегуляцияның арқасында дене температурасының тұрақтылығын сақтау болып табылады, яғни ағзаның қоршаған ортаға жылу беруді реттеу қабілеті. Микроклиматты нормалау

принципі-адам денесінің қоршаған ортамен жылу алмасуына оңтайлы жағдай жасау.

Есептеу техникасы маңызды жылу бөлу көзі болып табылады, ол температураның жоғарылауына және бөлмедегі салыстырмалы ылғалдылықтың төмендеуіне әкелуі мүмкін. Компьютерлер орнатылған үй-жайларда микроклиматтың белгілі бір параметрлері сақталуы тиіс. СН-245-71 санитарлық нормаларында қолайлы жағдай жасайтын микроклимат параметрлерінің шамалары орнатылған. Бұл нормалар жыл уақытына, еңбек процесінің сипатына және өндірістік үй-жайдың сипатына байланысты белгіленеді (4.1 кестеде көрсетілген).

Есептеу орталықтарының қызметкерлері орналасқан үй-жайлардың көлемі ауысымда бір мезгілде жұмыс істейтіндердің ең көп санын ескере отырып, 19, 5м³/адамнан кем болмауы тиіс. Компьютерлер орналасқан үй-жайларға таза ауаны беру нормалары 4.2 - кестеде келтірілген.

5.1 - кесте – Компьютерлер орнатылған үй-жайларға арналған микроклимат параметрлері

Жыл мезгілі	Микроклимат параметрлері	Шамасы
Суық	Үй-жайдағы ауа температурасы салыстырмалы ылғалдылық	22...24°C
	Ауа қозғалысының жылдамдығы	40...60% до 0,1м/с
Жылы	Үй-жайдағы ауа температурасы салыстырмалы ылғалдылық	23...25°C
	Ауа қозғалысының жылдамдығы	40...60% 0,1...0,2м/с

5.2 - кесте – Компьютерлер орналасқан үй-жайларға таза ауаны беру нормалары

Үй-жайдың сипаттамасы	Үй-жайға берілетін таза ауаның көлемдік шығыны, сағатына бір адамға м ³
Бір адамға 20м ³ дейін көлем	30-дан кем емес
Бір адамға 20 ... 40м ³	20-дан кем емес
Бір адамға 40м ³ артық	Табиғи желдету

Қолайлы жағдайларды қамтамасыз ету үшін ұйымдастыру әдістері (жыл және тәулік уақытына байланысты жұмыстарды жүргізуді ұтымды ұйымдастыру, еңбек және демалыс кезектесуі), сондай-ақ техникалық құралдар (желдету, ауаны баптау, жылыту жүйесі) қолданылады.

5.1.7 Жұмыс орнының эргономикасы

Ғылыми-техникалық прогресс, әртүрлі технологиялық процестерді механикаландыру және автоматтандыру адамның еңбек сипатының түбегейлі өзгеруіне әкеп соқты. Ол "адам - машина - өмір сүру ортасы" жүйелерінде басқару функциясын жүзеге асыратын оператор ретінде әрекет ете бастады. Адамға дене жүктемелері азайды, оның еңбек жағдайлары жақсарып, жарақат алу мүмкіндігі төмендеді. Алайда, бұл оның жұмысын толық жайлы етпеді, ал жүйенің өзі экологиялық қауіпсіз, өйткені бұл ретте жаңа, деп аталатын психофизиологиялық зиянды факторлар маңызды мәнге ие болды. Адамның психикалық қызметінің мәні айтарлықтай өсті және сонымен бірге – ТЖМСО жүйесін жетілдіру қажеттілігі. Сонымен қатар, барлық жүйе жұмысының сенімділігі мен тиімділігі елеулі түрде тәуелді болатын, оператордың қауіпсіз, авариясыз, жайлы еңбек жағдайларын қамтамасыз ету проблемасы маңызды болды.

Мұндай еңбек жағдайлары бірінші кезекте эргономикалық талаптар тұрғысынан программистің жұмыс орнын дұрыс ұйымдастырумен қамтамасыз етіледі.

Жұмыс орнын жобалау кезінде СанЕжН-да орнатылған талаптарды ескере отырып, жұмыс орнын таңдау керек. 2.2.2/2.4.1340-03.

ПЭВМ конструкциясы бейне дисплей терминалының (ВДТ) экранын фронтальды бақылауды қамтамасыз ету үшін берілген қалыпта бекітілген көлденең және тік жазықтықта корпусның бұрылу мүмкіндігін қамтамасыз етуі тиіс. ПЭВМ дизайны корпусы тыныш жұмсақ тонға бояуды қарастыру керек. ПЭВМ корпусы, пернетақта және басқа да блоктар мен ПЭВМ құрылғылары 0,4-0,6 шағылысу коэффициенті бар күңгірт беті болуы тиіс және жарықты жасауға қабілетті жылтыр бөлшектері болмауы тиіс.

ВДТ конструкциясы жарықтылық пен контрастылықты реттеуді көздеуі тиіс.

ЭЕМ пайдалануға арналған үй-жайларда қолданыстағы нормативтік құжаттама талаптарына сәйкес келетін табиғи және жасанды жарық болуы тиіс.

Сынаушының жұмыс үстелін ВДТ бүйір жағынан жарық ойықтарына бағытталатындай етіп орналастыру керек және табиғи жарық көбінесе солға құлап кетеді.

Бір жұмыс орнына жалпақ дискретті экрандар (сұйық кристалды, плазмалық) базасында ВДТ бар ПЭВМ қолданушыларының ауданы кемінде 6 м² құрауы тиіс. Қарастырылып отырған үй-жайдың ауданы-12 м².

Бейнемонитор экраны пайдаланушының көзінен 600-700 мм қашықтықта, бірақ алфавиттік-сандық белгілер мен символдардың мөлшерін ескере отырып, 500 мм-ден жақын болмауы тиіс.

Жұмыс үстелінің құрылымы қолданылатын жабдықтың жұмыс бетінде оның саны мен құрылымдық ерекшеліктерін, орындалатын жұмыстың сипатын ескере отырып оңтайлы орналасуын қамтамасыз етуі тиіс.

Бұл ретте эргономиканың заманауи талаптарына жауап беретін әр түрлі конструкциялы жұмыс үстелдерін пайдалануға жол беріледі. Жұмыс үстелінің бетінде 0,5 – 0,7 шағылысу коэффициенті болуы тиіс.

Жұмыс креслоларының конструкциясы ПЭВМ-де жұмыс істеу кезінде тиімді жұмыс қалпын ұстап тұруды қамтамасыз етуі, шаршаудың дамуының алдын алу үшін мойын-иық аймағы мен арқаның бұлшық етінің статикалық кернеуін төмендету мақсатында қалпын өзгертуге мүмкіндік беруі тиіс.

Жұмыс креслосы көтермелі-бұрылмалы, орындық пен Арқаның биіктігі мен көлбеу бұрыштары бойынша реттелетін, сондай-ақ Арқаның орындықтың алдыңғы шетінен арақашықтығы болуы тиіс, бұл ретте әрбір параметрді реттеу тәуелсіз, оңай жүзеге асырылатын және сенімді бекітілуі тиіс.

Орындықтың, арқаның және креслоның басқа элементтерінің беті жартылай жұмсақ, тайғанамайтын, әлсіз электрленетін және ауа өткізбейтін, ластанудан жеңіл тазартуды қамтамасыз ететін жабыны бар болуы тиіс.

Жұмыс бетінің биіктігі 680-800 мм шамасында реттелуі тиіс, мұндай мүмкіндік болмаған жағдайда үстелдің жұмыс бетінің биіктігі 725 мм құрауы тиіс.

Жұмыс үстелінің биіктігі 600 мм – ден кем емес, ені 500 мм – ден кем емес, тізе деңгейіндегі тереңдігі 450 мм – ден кем емес және созылған аяқ деңгейінде-650 мм-ден кем емес аяққа арналған кеңістігі болуы тиіс.

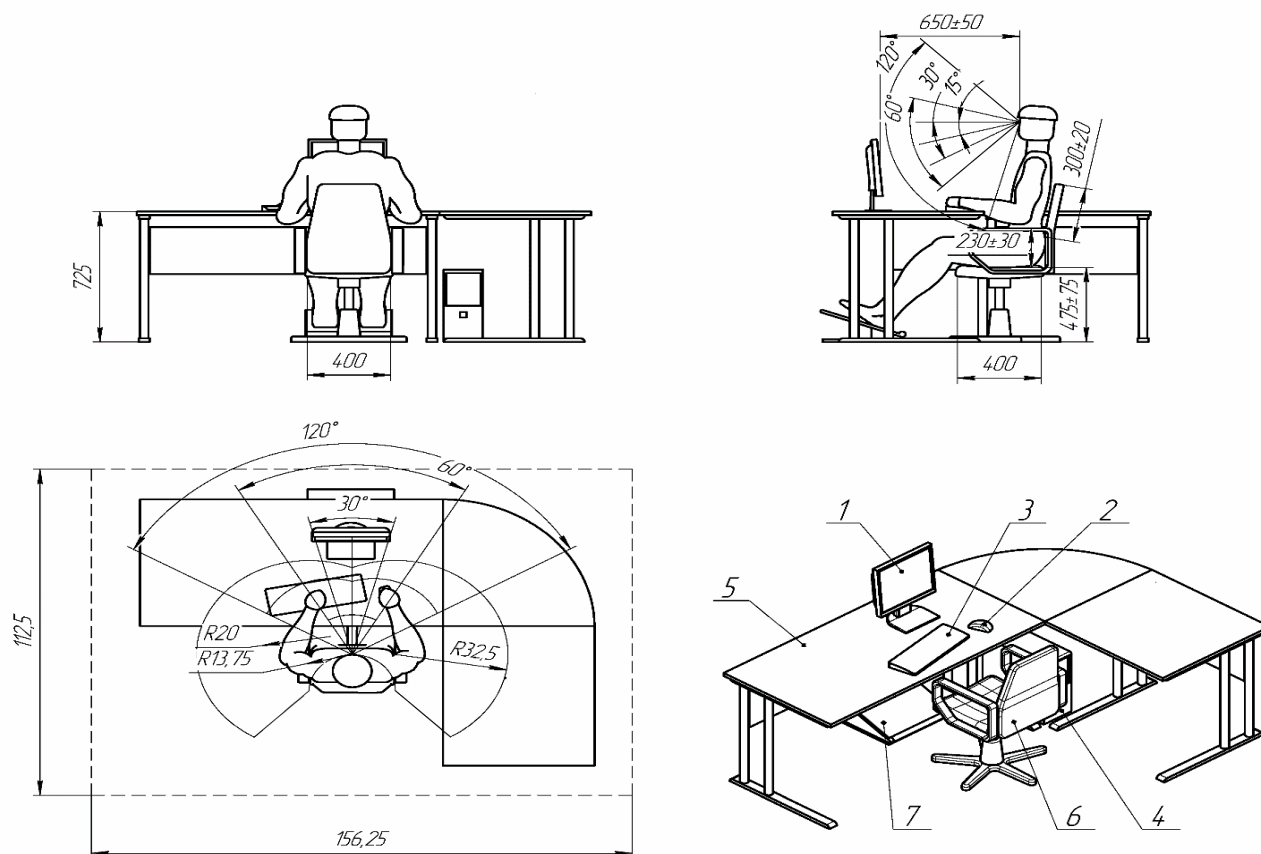
Жұмыс креслосының құрылымы:

- орындық бетінің ені мен тереңдігі кемінде 400 мм;
- алдыңғы шеті дөңгелектелген орындықтың беті;
- орындық бетінің биіктігін 400-550 мм шегінде және алға қарай көлбеу бұрыштарын 15 градқа дейін реттеу., және артқа 5 град дейін.;
- Арқаның тірек бетінің биіктігі 300 ± 20 мм, ені 380 мм-ден кем емес және көлденең жазықтықтың қисық радиусы-400 мм;
- ± 30 градус шегінде тік жазықтықта Арқалықтың еңіс бұрышы;
- орындық алдыңғы шетінен Арқаның қашықтығын 260-400 мм шегінде реттеу;
- ұзындығы кемінде 250 мм және ені 50-70 мм стационарлық немесе алмалы-салмалы шынтакшалар;
- шынтакшаларды отырғыштың үстіндегі биіктігі бойынша 230 ± 30 мм және шынтакшалар арасындағы ішкі қашықтықты 350-500 мм шегінде реттеу.

ПЭВМ пайдаланушының жұмыс орнын ені 300 мм кем емес, тереңдігі 400 мм кем емес аяққа арналған тіреуішпен, биіктігі бойынша 150 мм дейін және тіреуіш бетінің еңіс бұрышы бойынша 20 градусқа дейін реттеумен жабдықтау қажет. Тұғырықтың беті кедір - бұдырланған және алдыңғы жиегі бойынша биіктігі 10 мм борты болуы тиіс.

Клавиатураны үстелдің бетіне пайдаланушыға қараған шетінен 100 – 300 мм қашықтықта немесе негізгі столшадан бөлінген биіктігі бойынша реттелетін арнайы жұмыс бетіне орналастыру керек.

Көрсетілген талаптарға сәйкес программистің жұмыс орны 5.1 суретте жобаланды.

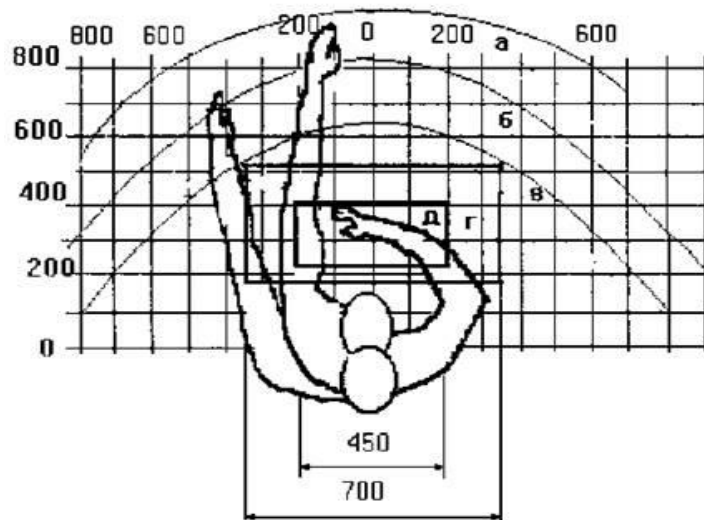


5.1 - сурет – Программистің эргономикалық жұмыс орнының сызбасы

мұнда 1 – монитор; 2 – тышқан; 3 – пернетақта; 4 – жүйелік блок; 5 – жұмыс орны; 6 – кресло; 7 – тұғырық;

Қолдардың барынша қол жетімділік аймағы - бұл иық буынында қозғалғанда қолмен барынша созылатын доғалармен шектелген жұмыс орнының моторлы алаңының бөлігі.

Оңтайлы аймақ-шынтақ буындарында шынтақ нүктесінде тіреуі бар және салыстырмалы қозғалмайтын иығы бар қозғалыс кезінде білеппен сипатталатын доғалармен шектелген жұмыс орнының моторлы алаңының бөлігі.



5.2 - сурет – Көлденең жазықтықтағы қолдардың қол жетімділік аймағы.

- а - барынша қол жеткізу аймағы;
- б - қол созылғанда саусақтардың қол жетімділік аймағы;
- в - алақанның жеңіл қол жеткізу аймағы;
- г - дөрекі қолмен жұмыс үшін оңтайлы кеңістік;
- д - жұқа қолмен жұмыс істеу үшін оңтайлы кеңістік.

Еңбек жағдайын жақсарту және өндірістегі жұмыс орындарын дұрыс эстетикалық безендіру Еңбекті жеңілдету үшін де, еңбек өнімділігіне оң әсер ететін оның тартымдылығын арттыру үшін де үлкен маңызға ие. Үй-жайлар мен жиһаздарды бояу көру, жақсы көңіл-күй үшін қолайлы жағдай жасауға ықпал етуі тиіс. Айтарлықтай нерв кернеуін және үлкен шоғырлануды талап ететін біркелкі ақыл - ой жұмысы орындалатын қызметтік үй-жайларда түсі тыныш реңктер-суық жасыл немесе көгілдір түстердің аз қанықпаған реңктері болуы тиіс.

Жұмыс орны - программист еңбек қызметін жүзеге асыратын және жұмыс уақытының көп бөлігін өткізетін кеңістіктің бір бөлігі. Инженердің еңбек қызметіне жақсы бейімделген жұмыс орны кеңістікке, формаға, өлшемге қатысты дұрыс және мақсатқа сай ұйымдастырылған, оған жұмыс кезінде ыңғайлы жағдайды және физикалық және психикалық шиеленіскен кезде жоғары еңбек өнімділігін қамтамасыз етеді.

Жұмыс орнын дұрыс ұйымдастыру кезінде инженердің Еңбек өнімділігі 8-20 пайызға өседі.

Осылайша, программист еңбегінің салауатты және қауіпсіз жағдайларын қамтамасыз ету проблемасын шешу - өндірістік қызмет процесінде аса маңызды міндеттердің бірі. Салауатты еңбек жағдайларын жасау кезінде компьютермен жұмыс істеу кезінде шаршау азаяды, өндірістік жарақаттану қаупі төмендейді. Демек, өндірістік процестің тиімділігі артады.

5.2 Жасанды жарықтандыруды нүктелік әдіспен есептеу

Шамдарды үй-жайда орналастыру келесі параметрлермен анықталады, м.

Бастапқы деректер:

$$A = 10 \text{ м;}$$

$$B = 8 \text{ м;}$$

$$H = 5 \text{ м;}$$

$$h_{pp} = 1,4 \text{ м;}$$

$$h_c = 0,5 \text{ м;}$$

$$h_p = H - h_c - h_{pp} = 5 - 1,4 - 0,5 = 3,1 \text{ м;}$$

мұнда H - үй-жайдың биіктігі;

h_c - шамдардың жабыннан қашықтығы;

h_{pp} - еденнен жұмыс бетінің биіктігі;

$h_p = H - h_c - h_{pp}$ - есептік биіктігі, шам мен жұмыс алаңының ара қашықтығы;

Жұмыс орнында қолайлы көру жағдайларын жасау үшін, жарық көздерінің соқыр әсерімен күресу үшін шамдардың еденнен ең аз биіктігін шектеу талаптары енгізілген.

L - көрші шамдардың немесе қатарлардың арасындағы қашықтық (егер (A) және ені (B) бойынша үй – жай қашықтығы әртүрлі, онда олар LA және LB белгіленеді), 1-шеткі шамдардан немесе қатардан қабырғаға дейінгі қашықтық.

Шамдардың шеткі қатарынан қабырғаға дейінгі оңтайлы қашықтық l (0,3-0,5) L тең қабылдау ұсынылады.

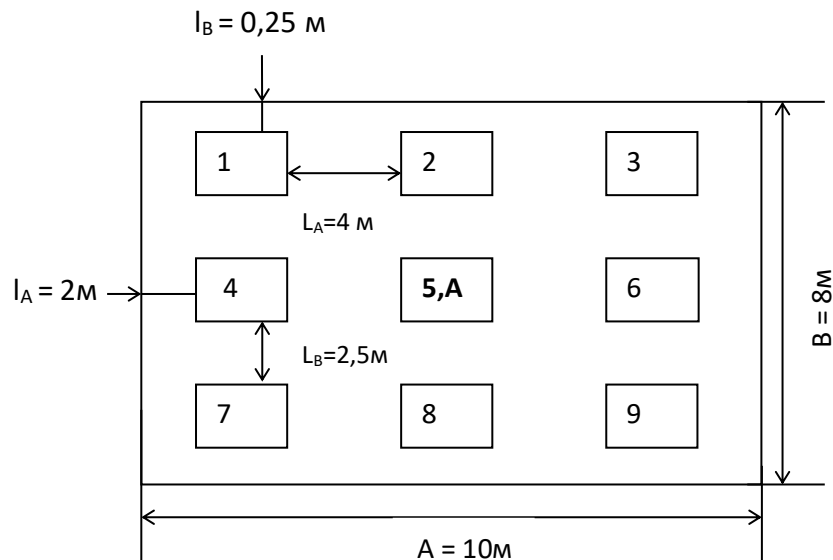
Шамдардың тиімді орналасуының интегралды өлшемі $\lambda = L / h$ шамасы болып табылады, оның азаюы құрылғы мен жарықтандыруға қызмет көрсету қымбаттайды, ал шамадан тыс ұлғаюы жарықтандырудың біркелкі еместігіне әкеледі. Әр түрлі шамдарға λ мәндері үшін 4.1 кесте көрсетілген.

5.1 - кесте – Шамдардың әртүрлі типтері үшін λ мәндері.

Типтік қисық	λ мәні	
	Ұсынылатын	Ең көп рұқсат етілген
Қойылтылған (К)	0,4 - 0,7	0,9
Терең (Г)	0,8 - 1,2	1,4
Косинусты (Д)	1,2 - 1,6	2,1
Біркелкі (М)	1,8 - 2,6	3,4
Жартылай қалың (Л)	1,4 - 2,0	2,3

L шамдардың арасындағы қашықтық $L = \lambda * h_p$ ретінде анықталады.

Шамдардың орналасу схемасы 5.1 суретте көрсетілген.



5.3 - сурет – Шамдардың орналасу схемасы

Қашықтықты есептейміз немесе оны логикалық ойлап анықтаймыз:

Біз 10 м бөлменің ұзындығы, яғни шамдарды олардың арасында 4 м болатындай етіп орналастырамыз, ал қабырғадан шамға дейінгі қашықтық 1 м болатындай етіп орналастырамыз. Демек, $((10 - (4 + 4)) / 2 = 1)$.

Сондай – ақ ені бойынша - ені 8 м, ал шамдардың арасындағы қашықтық 2,5 м, ал қабырғадан 0,25 м. Демек, $((8 - (2,5 + 2,5 + 2,5)) / 2 = 0,25)$.

Есептеу ыңғайлылығы үшін шамдарды нөмірледік.

Ең алдымен ортадағы негізгі шамды есептейміз, ол бізде 5 – ші нөмерлі шам. Және келесі басқа шамдар осы шамнан ара қашықтығынан есептелінеді.

Төбенің және тік d арасындағы бұрышты анықтаймыз, бұл $tg\alpha$ шамның берілген биіктігі бойынша есептелінеді:

$$tg\alpha = \frac{d}{h_p} \tag{5.1}$$

мұнда d шамның осі проекциясынан есептік нүктеге дейінгі қашықтық (d шамасы жоспар бойынша өлшенеді), М;

$$h_p = H - h_c - h_{pn} = 5 - 0,4 - 1,2 = 3,4 \text{ м};$$

$$tg\alpha = \frac{1,2}{3,4} = 0,35 \tag{5.2}$$

Енді интернеттен Брадис кестесін қолданамыз.
Егер 0,35 болса, онда бұрыш тең болады: 19° .
Сонда $\cos 19^\circ = 0,94$ тең болады.

5.2 - кесте – Тангенстің градус бойынша мәні.

0°-ден 360° -қа дейінгі бұрыштар тангенсының толық кестесі	
Бұрыш градуспен0	tg (тангенс)
0°	0
1°	0,0775
35°	0,7002
36°	0,7265
37°	0,7536
38°	0,7813
39°	0,8098
40°	0,8391
58°	1,6003
59°	1,6643
60°	1,7321

$$e_1 = \frac{132 * \cos^3(19)}{3.4^2} = \frac{132*0,94}{11.56} = 10,73 \text{ лк}; \quad (5.3)$$

$I_a = 139$ кд тең 15° болған жағдайда; (5.3 - кесте бойынша)

Және де тангенс мәндерін арктангенске келтіру үшін 4.2 – кестеге қараймыз.

5.3 - кесте – Жарық күші I_ξ кд орналасу бұрышы ξ бойынша.

Светильник түрі	Жарық күші I_ξ кд орналасу бұрышы ξ бойынша										
	0	5	15	25	35	45	55	65	75	85	90
ПВЛМ-1x40	139	135	132	115	104	84	63	44	22	6	0

Келесі шамдар 2 мен 8 шамды аламыз олардын ара қашықтығы үстелден бірдей болғандықтан оларды бірге есептейміз.

2 және 8 шамдар үшін:

$$d_2 = 2,5 \text{ м}; \quad (5.4)$$

$$\alpha = \text{atan}(2,5/3,4) = \text{atan}(0.74) = 36,5^\circ; \quad (4.2 - \text{кесте бойынша})$$

Егер тангенс 0,74 болса, онда бұрыш тең болады: $36,5^\circ$. Сонда $\cos^3 36,5 = 0,37$ тең болады.

$$e_2 = \frac{108,2 * \cos^3(36,5)}{3,4^2} = \frac{108,2 * 0,37}{11,56} = 3,46 \text{ лк}; \quad (5.5)$$

$I_a = 108,2$ кд тең $36,5^\circ$ болған жағдайда; (4.3 – кесте бойынша)

Келесі шам 4 және 6 олардың да ара қашықтығы бірдей болғандықтан оны да бірге есептейміз.

4 пен 6 шамдар үшін:

$$d_3 = 4 \text{ м}; \quad (5.6)$$

$$\alpha = \text{atan}(4/3,4) = \text{atan}(1,18) = 49,7^\circ; \quad (4.2 – \text{кесте бойынша})$$

Егер тангенс 1,18 болса, онда бұрыш тең болады: $49,7^\circ$. Сонда $\cos^3 49,7 = 0,27$ тең болады.

$$e_3 = \frac{74,4 * \cos^3(49,7)}{3,4^2} = \frac{74,4 * 0,27}{11,56} = 1,74 \text{ лк}; \quad (5.7)$$

$I_a = 74,4$ кд тең $49,7^\circ$ болған жағдайда; (4.3 кесте бойынша)

Енді 1,3,7,9 шамдар бірдей қашықтықта бірдей бұрышпен тұрғандықтан оларды бірге есептейміз. Ен алдымен олардың ара қашықтығын тауып аламыз.

1,3,7,9 шамдар үшін:

$$d_4 = \sqrt{4^2 + 2,5^2} = 4,72 \text{ м}; \quad (5.8)$$

Енді бұрышын есептейміз:

$$\alpha = \text{atan}(4,72/3,4) = \text{atan}(1,39) = 53^\circ; \quad (4.2 – \text{кесте бойынша})$$

Егер тангенс 1,39 болса, онда бұрыш тең болады: 53° . Сонда $\cos^3 53 = 0,6$ тең болады.

$$e_4 = \frac{60 * \cos^3(53)}{3,4^2} = \frac{60 * 0,6}{11,56} = 3,1 \text{ лк}; \quad (5.9)$$

$I_a=60$ кд тең 53^0 болған жағдайда; (4.3 – кесте бойынша).

Жарықтандырудың жиынтық шарты тең болады:

$$\sum E = e_1 \cdot 1 + e_2 \cdot 2 + e_3 \cdot 2 + e_4 \cdot 4 = 10,7 \cdot 1 + 3,46 \cdot 2 + 1,74 \cdot 2 + 3,1 \cdot 4 = 33,5 \quad (5.10)$$

Барлық деректерді қойып мына формуламен анықтаймыз:

$$E_{AG} = \frac{\mu \cdot F_l}{1000 \cdot K_3} \sum_1^n e_{AGn}, \quad (5.11)$$

мұнда μ -қашықтағы шамдардан жарықтандыруды және қабырғалардан, төбеден және есептік бетінен шағылысқан жарық ағынын ескеретін коэффициент. Бұл коэффициент шамдардың қуатын арттыруға жол бермеу үшін түзету ретінде енгізіледі. Тікелей жарық эмальданған шамдарда $\mu=1,1-1,2$. Айналы кезде $\mu=1,0$. Тікелей жарық шамдарында $\mu=1,3-1,6$.

- Біз $\mu=1,1$ таңдаймыз.
- Жарық ағыны TL-D 58W / 865 типті лампаның $F_l = 5000$ Лм
- K_3 - қор коэффициенті, $K_3 = 1,2$ (4.4 кесте бойынша).

5.4 - кесте – Қор коэффициентінің мәні

Үй-жайлар мен аумақтар	Үй-жайлар мысалдары	Қор коэффициенті K_3				
		ЕО және жарық өткізу орналасуы кезінде материал			Жасанды жарықтандыру кезінде	
		Тігінен	Көлбеу	Көлденең	Газразрядт	Қыздыру
1	2	3	4	5	6	7
1. Жұмыс аймағында бар ауа ортасы бар өндірістік үй-жайлар: а) 5мг/м ³ шаң, түгін, тұяқтар	Агломерациялық фабрикалар, цемент зауыттары және құю цехтарының шабу бөлімшелері. Ұсталық, құю, дәнекерлеу, Құрастырмалы темір бетон цехтары.	1,5	1,7	2	2	1,7
б) 1-ден 5 мг/ м ³ дейін шаң, түгін, тұяқтар;	Құрал-саймандық, құрастыру, Механикалық, механикалық жинау, тігу цехтары.	1,4	1,5	1,8	1,8	1,5

5.4-кестенің жалғасы

в) 1 мг/м ³ кем шаң, түтін, тұяқтар;	Қышқылдар, сілтілер, күйдіргіш химиялық реактивтер, улы химикаттар, тыңайтқыштар өндіру жөніндегі химия зауыттарының цехтары.	1,3	1,4	1,5	1,5	1,3
г) ылғалмен жанасу кезінде қышқылдардың әлсіз ерітінділерін түзуге қабілетті булардың, қышқылдардың, сілтілердің, газдардың едәуір шоғырлануы.	Гальваникалық жабындар және гальванопластика цехтары	1,5	1,7	2	1,8	1,5
2. Ауа тазалығы бойынша ерекше режимдегі Үй-жайлар: а) техникалық қабаттан б) үй-жайлардан төменнен		-	-	-	1,3 1,4	1, 1 6 1, 2
3. Қоғамдық және тұрғын үй ғимараттарының үй-жайлары.	Кабинеттер, оқу үй-жайлары, зертханалар, сауда залдары және т. б.	1,2	1,4	1,5	1,5	1, 3

Барлық деректерді формулаға қоямыз:

$$E_{AG} = \frac{5000 \cdot 2 \cdot 1,1 \cdot 33,5}{1000 \cdot 1,2} = 307,8 \text{лк} \quad (5.12)$$

Ал норма бойынша 200лк болу қажет. Демек, өндірістік жайды жарықтандыру жүйесін қайта құру қажет емес.

$$E_{AG} > E_n \quad (5.13)$$

Нормаланған жарықтандыруды қамтамасыз етеді.

Кәсіпорында дұрыс жобаланған және орындалған жарықтандыру қалыпты өндірістік қызмет мүмкіндігін қамтамасыз етеді.

Қорытынды

21 ғасыр – ақпараттық технологиялар ғасыры. Бүкіл әлемде адамдар жаңа технологияларды ұстануда. Көбінесе адамдарға үйден шықпай-ақ кез-келген операцияны сатып алуға, сатып алуға, жасауға ыңғайлы және тиімдірек болады.

Сондай тиімділіктердің бірі ол - интернет дүкені. Интернет-дүкеннің пайда болуымен көптеген адамдардың өмірі жеңілдеді, өйткені әлемнің түкпір-түкпірінен тауарларға тапсырыс беру мүмкіндігі пайда болды. Бұл қазіргі әлемнің жаңашыл бағдарламашыларының еңбегі. Егер бұрын интернет-дүкен жарамдылық мерзімі жоқ тауарларды сату үшін пайдаланылған болса, қазір интернет-дүкендер арқылы сіз кез-келген нәрсеге тапсырыс бере аласыз. Клиенттермен тиімді қарым-қатынас орнату және бәсекеге қабілеттілікті арттыру көбінесе инновациялық технологияларды енгізуге байланысты. Қазіргі жағдайда интернет-дүкеннің тиімділігі мен сәттілігі нарықтағы жағдайдың өзгеруіне байланысты. Бірақ бұл тек тауарларды ұсынумен аяқталмайды, іс жүзінде интернет-дүкеннің идеясы жан-жақты: ол соңғы тұтынушы үшін ең қолайлы жағдай жасауды қарастырады.

Дипломдық жұмысты орындау барысында “Әйелдер киімдеріне онлайн тапсырыс ұйымдастыруға арналған веб-сайт әзірлеу” тақырыбына байланысты Python бағдарламалау тілінің Django фреймворкы арқылы веб-сайт құрылды.

Интернет-дүкен тауарлардың әртүрлі санаттарын ұсынады. Бұл дипломдық жобада MVC технология қолданылады. IDE интеграцияланған даму ортасы (Integrated Development Environment) PyCharm және SQLite деректер қорын басқару жүйесі осы жобада қолданылады. Әзірлеу кезінде Python программалау тілінің Django фреймворкы пайдаланылды. Бұл бізге статистикалық және динамикалық мазмұнды жасауға мүмкіндік береді. Сондай-ақ, біздің веб-қосымшаның логикасына жауап беретін модельдер қолданылды. Веб сайт екі бөліктен тұрады – серверлік және клиенттік.

Веб-сайт барлық заманауи технологияларды пайдалана отырып жасалды. Адамдардың өмірін жеңілдету мақсатында барынша ыңғайлы болатын веб-сайт. Түсінікті әрі функционалды. Қазіргі еліміздегі карантин жағдайына байланысты бұл сайт өте қажет және адамдар үшін өте қолайлы. Себебі осы веб-сайт арқылы үйден шықпайда онлайн тапсырыс беруге болады.

Веб-сайтты жасау үшін барлық заманауи, қажетті және программаны барынша оңтайландыратын технологиялар қолданды. Ол технологиялар веб-сайттың жылдамдығын әлдеқайда арттырып, қолданушыларға еш қолайсыздық туғызбайды. Программа барлық стандарттарға сай жасалды.

Әдебиеттер тізімі

1 Cameron Davidson-Pilon. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference // Addison-Wesley Data & Analytics, 2015.

2 UML – диаграмма классов // Сайттың электрондық нұсқасы
<https://prog-cpp.ru/uml-classes/>

3 Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.

4 Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

5 Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.

6 Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.

7 Django – Python framework // Сайттың электрондық нұсқасы
<https://realpython.com/django-setup/>

8 З.Д. Еркешева, Г.Ш. Боканова, 5В070400 – Есептеу техникасы және бағдарламамен қамтамасыз ету мамандығының студенттері үшін диплом жұмысының экономика бөлімін орындауға әдістемелік нұсқаулар. – Алматы: АЭЖБУ, 2014 – 27 с.

9 Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальностей 5В070400 – Вычислительная техника и программное обеспечение, 5В070300 – «Информационные системы», 5В060200 – «Информатика». – Алматы: АУЭС, 2019 – 33 с.

10 Sulpak дүкенінің URL- адресі:
https://www.sulpak.kz/p/kompyuternaya_tehnika

А қосымшасы (міндетті)

А.1 Кіріспе

Дипломдық жобаның негізгі идеясы Әйелдер киімдеріне онлайн тапсырыс ұйымдастыруға арналған веб-сайт әзірлеу.

Идеаны жүзеге асыру барысында заманауи технологиялар қолдану есепке алынды. Дипломдық жобаның мақсаты – Әйелдер киімдеріне онлайн тапсырыс ұйымдастыруға арналған веб-сайт әзірлеу.

А.1.1 Өңдеудің мақсаты мен қызметі

Дипломдық жобаның мақсаты: Әйелдер киімдеріне онлайн тапсырыс ұйымдастыруға арналған веб-сайт әзірлеу

Дипломдық жобаның өзектілігі – адамдардың сауда жасауын оңтайландыру, тапсырмаларды жоспарлауға мүмкіндік берілуі және олардың орындалуын бақылау. Адамдар мен сауда жұмысын жеңілдету. Тірі кезекте тұрмас үшін онлайн түрде тапсырыс беру. Сонымен қатар қазіргі карантин уақытында осындай сервистердің өзектілігі өте жоғары.

Дипломдық жобаның есептік қойылымы

Құрылатын жүйеде қолданушымен мен әкімші қызметкерлерге қатысты барлық ақпараттар деректер қорында сақталатын жүйе құру.

- а) жүйе құрылымын құру;
- б) бағдарламалау тілін таңдау;
- в) деректер қорын басқару жүйесін таңдау;
- г) тиімді және түсінікті программалық интерфейсті құру;
- д) қосымша бойынша ақпарат алуды ұйымдастыру;

Дипломдық жобаның нысаны

Қолданушыларының жұмысын жеңілдетуге арналған веб-сайт.

Дипломдық жобаның міндеттері

- а) құжаттар мен есептерді автоматтандыру;
- б) командада жұмысты ұйымдастыру;
- в) тапсырмалардың уақытылы орындалуын қадағалап, жұмыстың болмай қалмауын және сәтсіздікке ұшырамауына мүмкіндік бермеу.

А.1.2 Қолдану саласы

Бұл жүйе арқылы стоматологиялық клиникалар мен адамдар арасындағы қарым-қатынас көрсетіледі және қызмет сапасын жоғарылатуға мүмкіндік береді.

А қосымшасының жалғасы

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
UI	Қолданушы интерфейсі (User Interface)
SciPy	Sigh Pie
Uml	Unified Modeling Language

А.2 Жалпы сипаттамасы

Құрылатын жүйе арқылы клиенттерден келіп түскен тапсырмаларды басшының жұмысшыларына уақытылы жіберіп және сол жұмыстарды қабылдап алу, және клиенттерге дайын жұмыстарды жіберіп отыру.

Дипломдық жұмыстың нәтижесінде келесі негізгі тапсырмалар орындалуы қажет:

- веб-сайттың құрылымын құру;
- тиімді және түсінікті веб-сайт интерфейсін құру;
- компания қызметкерлері мен клиенттер арасындағы әрекеттерін ұйымдастыру;
- қосымша бойынша ақпарат алуды ұйымдастыру.

А.2.1 Пайдаланушы интерфейстер

Ақпараттарды алу үшін басты бетте қолданушы түрін таңдап авторизация жасау қажет.

Қолданушының екі түрі бар:

- басшы;
- қызметкер.

Басшы сәтті авторизация өткеннен кейін ол өзінің мәзірінде толық қолданушыларын және тек өз бөліміндегі қолданушыларды көре алады. Дәл осы басшыға клиенттерден тапсырмалар келіп түседі және сол тапсырмаларды басшы өз қолданушыларына белгілі уақытта тапсырылу керек деп жібереді, сол жұмыстарды қабылдап алады. Қолданушыға жаңалықтар беті тапсырмалар келіп сол жұмысты атқарады.

А.2.2 Аппараттық интерфейстер

PyCharm мен жұмыс жасау үшін:

- процессор Intel® Core i3 және жоғары немесе 4 ГГц аналогтық процессор AMD® технологиясы бойынша SSE2;
- 2 Гб жедел жады (8 Гб болуы ұсынылады);
- 2 Гб қатты диск жадында бос орын болу керек;
- видео картасы 512Мб- тан кем болмауы (1 Гб және одан көп болуы ұсынылады);
- тінтуір – PS/2 стандартты;
- пернетақта (орысша/қазақша);
- монитор (көрсету қабілеті) – 17’’(1024x768).

А.2.3 Программалық интерфейстер

Бағдарламалық қамтамасыздандыруды құру:

- а) Microsoft Windows, Mac OS X, Linux операциялық жүйелері;
- б) Python;

А.2.4 Адаптация бойынша талаптар

Программалық кешен Windows 7, Windows 8, Windows 10, Mac OS X, Linux платформаларында жұмыс істеуі тиіс, бұл немесе басқа платформаларды қолдануда қажетті программалық компоненттеріңғайлы орнатылуы керек

А.3 Өңдеуге талаптар

А.3.1 Функциональдық талаптар

Қызметкер немесе басшы жеке кабинетіне кіру үшін өзінің логин мен құпия сөзін енгізу керек. Қосымшаға кіргеннен кейін өзіне қажетті ақпараттарды көре алады.

А.3.1.2 Функциональды талаптарды ұйымдастыру

Жүйе клиент-сервер архитектурасының бөлігі болып табылады. Клиент пен сервер арасындағы байланыс транспортты-желі деңгейінде TCP/IP хаттамасының көмегімен жүзеге асады. Ақпараттық алмасуды қолданбалы деңгейдегі HTTP хаттамасы және оның HTTPS кеңейтілімі қолданылады.

А.3.2.1 Өнімділігі

А қосымшасының жалғасы

Қосымшаның өнімділігін арттыру үшін компьютер Wi-Fi желісінде немесе локалды интернетте болуы керек.

А.3.2.2 Қатынау және сенімділік

Жүйе жұмысы жұмыс уақытының аралығында тоқтап қалмауы тиіс:

- профилактикалық жұмысты белгілі бір уақытта жүргізу керек;
- авариялық жағдайларды болдыртпау.

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

А.3.2.3 Қатені өңдеу

Жүйедегі барлық қателер өңделуі тиіс және қателер туралы хабарлар пайдаланушылар қабылдай алатын түрде көрсетілуі керек.

Хабарлау жүйесінде пайдаланушылардың мүмкін емес әрекеттерін, ақпарат енгізудің қате үлгілерін іске асыру керек.

А.3.2.4 Интерфейсті талаптар

Қосымша жылдам ақпарат алу үшін қолданушыға түсінікті және ыңғайлы интерфейс болуы қажет. Қосымшаның интерфейсін жалпы қабылданған стандарттардан ауытқымауы қажет.

А.3.2.5 Шектеулер

Қолданылатын платформалар: Кез-келген интернетке мүмкіндігі бар компьютерлерде жұмыс істейді.

А.3.3 Кері талаптар

Процессор типі – Intel core i3 3 ГГц немесе жоғары. Осы талап қанағаттандырылмаса нәтижеге жету мүмкін емес.

Керісінше талаптар бұл талаптардың спецификациясы. Онда қосымша нені жасамауы жайлы анықтайды. Керісінше талаптардың шексіз саны бар болуы логикалы: мүмкін болатын түсінбеушілікті түсіндіретінін ғана белгілеу қажет.

Ә қосымшасы

Бағдарлама мәтіні

Models.py файлы

```
from django.db import models
from django.urls import reverse
from parler.models import TranslatableModel, TranslatedFields

class Category(TranslatableModel):
    translations = TranslatedFields(
        name = models.CharField(max_length=200,
                                db_index=True),
        slug = models.SlugField(max_length=200,
                                db_index=True,
                                unique=True)
    )

    class Meta:
        # ordering = ('name',)
        verbose_name = 'category'
        verbose_name_plural = 'categories'

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('shop:product_list_by_category',
                      args=[self.slug])

class Product(TranslatableModel):
    translations = TranslatedFields(
        name = models.CharField(max_length=200, db_index=True),
        slug = models.SlugField(max_length=200, db_index=True),
        description = models.TextField(blank=True)
    )
    category = models.ForeignKey(Category,
                                related_name='products',
                                on_delete=models.CASCADE)
    image = models.ImageField(upload_to='products/%Y/%m/%d',
                              blank=True)
```

Ә қосымшасының жалғасы

```
price = models.DecimalField(max_digits=10, decimal_places=2)
available = models.BooleanField(default=True)
created = models.DateTimeField(auto_now_add=True)
updated = models.DateTimeField(auto_now=True)
```

```
#class Meta:
# ordering = ('name',)
# index_together = (('id', 'slug',))

def __str__(self):
    return self.name

def get_absolute_url(self):
    return reverse('shop:product_detail',
                  args=[self.id, self.slug])
```

views.py файлы

```
from django.shortcuts import render, get_object_or_404
from .models import Category, Product
from cart.forms import CartAddProductForm

def product_list(request, category_slug=None):
    category = None
    categories = Category.objects.all()
    products = Product.objects.filter(available=True)
    if category_slug:
        language = request.LANGUAGE_CODE
        category = get_object_or_404(Category,
                                     translations__language_code=language,
                                     translations__slug=category_slug)
        products = products.filter(category=category)
    return render(request,
                  'shop/product/list.html',
                  {'category': category,
                  'categories': categories,
                  'products': products})

def product_detail(request, id, slug):
    language = request.LANGUAGE_CODE
```

Ә қосымшасының жалғасы

```
product = get_object_or_404(Product,
                             id=id,
                             translations__language_code=language,
                             translations__slug=slug,
                             available=True)
cart_product_form = CartAddProductForm()

return render(request,
              'shop/product/detail.html',
              {'product': product,
              'cart_product_form': cart_product_form})
```

Base.html файлы

```
{% load i18n %}
{% load static %}
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>{% block title %}{% trans "Shop.kz" %}{% endblock %}</title>
  <link href="{% static "css/base.css" %}" rel="stylesheet">
</head>
<body>
<div id="header">
  <a href="/" class="logo">{% trans "Shop.kz" %}</a>
  {% get_current_language as LANGUAGE_CODE %}
  {% get_available_languages as LANGUAGES %}
  {% get_language_info_list for LANGUAGES as languages %}
  <div class="languages">
    <p>{% trans "Language" %}:</p>
    <ul class="languages">
      {% for language in languages %}
        <li>
          <a href="/{ language.code }/"
            class="selected"{% endif %}>
            {{ language.name_local }}
          </a>
        </li>
      {% endfor %}
    </ul>
```

```

</div>
  <a id="loginid" style="margin-left: 60%; margin-top: 25%"
href="/admin">{% trans "Login" %}</a>
</div>
<div id="subheader">
  <div class="cart">
    {% with total_items=cart|length %}
      {% if cart|length > 0 %}
        {% trans "Your cart" %}:
        <a href="{% url "cart:cart_detail" %}">
          {% blocktrans with total_items_plural=total_items|pluralize
total_price=cart.get_total_price %}
            {{ total_items }} item{{ total_items_plural }},
            {{ total_price }} ₸
          {% endblocktrans %}
        </a>
      {% else %}
        {% trans "Your cart is empty." %}
      {% endif %}
    {% endwith %}
  </div>
</div>
<div id="content">
  {% block content %}
  {% endblock %}
</div>
</body>
</html>

```

List.html файлы

```

{% extends "shop/base.html" %}
{% load i18n %}
{% load static %}

{% block title %}
  {% if category %}{% blocktrans %}{{ category.name }}{% endblocktrans
%}
  {% else %}{% trans "Products" %}{% endif %}
{% endblock %}

{% block content %}

```

```

<div id="sidebar">
  <h3>{% trans "Categories" %}</h3>
  <ul>
    <li {% if not category % }class="selected"{% endif %}>
      <a href="{% url "shop:product_list" %}">{% trans "All" %}</a>
    </li>
    {% for c in categories % }
      <li {% if category.slug == c.slug % }class="selected"{% endif %}>
        <a href="{ { c.get_absolute_url } }">{{ c.name }}</a>
      </li>
    {% endfor % }
  </ul>
</div>
<div id="main" class="product-list">
  <h1>{% if category % }{% blocktrans % }{{ category.name }}{%
endblocktrans % }{% else % }{% trans "Products" % }{% endif %}</h1>
  {% for product in products % }
    <div class="item">
      <a href="{ { product.get_absolute_url } }">
        
      </a>
      <a href="{ { product.get_absolute_url } }">{ { product.name } }</a><br>
        { { product.price } } ₸
    </div>
  {% endfor % }
</div>
{% endblock % }

```

Detail.html файлы

```

{% extends "shop/base.html" % }
{% load i18n % }
{% load static % }

{% block title % }
  { { product.name } }
{% endblock % }

{% block content % }
  <div class="product-detail">

```


Ә қосымшасының жалғасы

```

<h1>{{ product.name }}</h1>
<h2><a href="{{ product.category.get_absolute_url }}">{{
product.category }}</a></h2>
<p class="price">{{ product.price }} ₸</p>

<form action="{% url "cart:cart_add" product.id %" method="post">
  {{ cart_product_form }}
  {% csrf_token %}
  <input type="submit" value="{% trans "Add to cart" %}">
</form>
  {{ product.description|linebreaks }}
</div>
{% endblock %}
```