

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі

(аты-жөні, ғылыми дәрежесі, атағы)

« _____ » 20__ ж.

(колы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Фирманың Тауар аймағының есепке
шығуы

мамандығы бойынша

Орындаған Моңғұржанов Бексүтжан ПОК-12-1

(аты - жөні)

(тобы)

Жетекші

(аты-жөні, ғылыми дәрежесі, атағы)

Кенесшілер :

Экономикалық бөлім бойынша :

Э.Э.К. доцент Түзелбаев Б.С.

(ғылыми дәрежесі, атағы, аты-жөні)

« 30 » 05 2016 ж.

(колы)

Өмір тіршілігі қауіпсіздігі бойынша:

Тех. ғылымдар докторы, профессор Хакимжанов Т.Е.

(ғылыми дәрежесі, атағы, аты-жөні)

« 02 » 06 2016 ж.

(колы)

Есептеу техникасын қолдану бойынша :

Инж.ғ.ғ. профессор Уманбетов М.М.

(ғылыми дәрежесі, атағы, аты-жөні)

« 03 » 06 2016 ж.

(колы)

Мөлшер бақылаушы:

Инж.ғ.ғ. профессор Урманбетова М.М.

(ғылыми дәрежесі, атағы, аты-жөні)

« 03 » 06 2016 ж.

(колы)

Пікір жазушы :

Әлеуметтік менеджмент маманы Ержан Абрахманович

(ғылыми дәрежесі, атағы, аты-жөні)

« 03 » 06 2016 ж.

(колы)

Алматы 2016

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

"Ақпараттық және аэротранспорт технологиялары факультеті
"Есептеу техникасы және бағдарламалық қамтамасыз ету" мамандығы
"Компьютерлік технологиялар" кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Шидуржанов Әлқашым Ахметұлы
(аты - жөні)

Жоба тақырыбы Құрмалы тілдік әдістермен есептеу

ректордың «19» қазан 2015 ж. №148 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: « » 20 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері)

Байланыс мақсатына сәйкес құрмалық жобаны құрмалық тілдік әдістермен есептеу және құрмалық мақсаттың өздерге қаратылған веб-сайтты және жобаның іске асырылуы.
Дипломның мақсатына сәйкес құрмалық мақсаттың іске асырылуы және құрмалық деректерінің қорықталуы.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

Мәліметтің мақсатына сәйкес құрмалық мақсаттың мақсатына сәйкес құрмалық мақсаттың іске асырылуы.

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Модель жасауға байланысты әрқайсысының маңызы	02.04.16	
2	Ғылыми жұмыстың теориялық зерттеулері	12.04.16	
3	Деректер жарнақ жазу	20.04.16	
4	Программалау	01.05.16	
5	Мәтіндік және дыбыстық	08.05.16	
6	Мұқият зерттеу және мод. әдістерін қолдану	10.05.16	
7	Мұқият байқау және сынақтар және үлгілер	11.05.16	
8	Әрбір үлгінің мұқият зерттеуі	15.05.16	
9	Иллюстрациялар және суреттер	13.05.16	
10	Мұқият және үлгілердің сыныптық бағалауы	13.05.16	
11	Мәтіндік құрылымдарды құрастыру және мәтіндік беруді және мәтіндік маңызы	14.05.16	
12	Сервистің программалауы	14.05.16	
13	Техникалық бағалау		

Тапсырманың берілген уақыты « _____ » _____ 20 _____ ж.

Кафедра меңгерушісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Жоба жетекшісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Орындалатын тапсырманы қабылдаған студент

(қолы)

(аты-жөні)

Аңдатпа

Қойылған тапсырмаға сәйкес дипломдық жобада фирманың тауар айналымын есепке алуға және тұтынушының тапсырысын өңдеуге арналған веб-бағдарлама жобаланып, іске асырылды.

Дипломдық жобада алынған нәтиженің фирманың тауар айналымын есепке алуға жұмсалатын уақыт пен қаржыны барынша азайтуға және оңтайландыруға мүмкіндік беретіндігі көрсетілді.

Бағдарлама интерфейсі web-негізделген технологияларды пайдалана отырып әзірленген және World Wide Web Consortium талаптарына жауап береді. Жобаның бизнес-логикасы серверлік скрипті бағдарламалау тілінде, халықаралық OWASP қауіпсіздік стандартына сәйкес іске асырылды.

Аннотация

В дипломном проекте согласно поставленным задачам было разработано и реализовано веб–приложение предназначенное для учета товарооборота фирмы обработки заявок от клиентов.

Было показано, что полученные результаты оптимизируют время и ресурсы нужные для учета товарооборота.

Интерфейс приложения было разработано согласно стандартам World Wide Web Consortium. Бизнес-логика проекта было разработана серверным языком программирования согласно требованиям безопасности OWASP.

Abstract

At the graduation according to the tasks it was designed and implemented a web-based application designed to account for claims processing company turnover from clients.

It has been shown that the results obtained to optimize the time and resources necessary to take account of turnover.

It was developed according standards World Wide Web Consortium interface applications. Business login of the project was developed by the server programming language according to the safety requirements of the OWASP.

Мазмұны

Кіріспе	8
1 Жоба тақырыбы бойынша аналитикалық талдау	9
1.1 Тасымалдауды ұйымдастыру негіздері.....	9
1.2 Жүк және жолаушы ағындарының сипаттамалары.....	11
1.3 Жүк ағындарының сипаттамасы.....	12
1.4 Жүкті тасымалдауға қабылдау.....	12
1.5 Желілік программалаудың транспорттық тапсырмасының берілуі және математикалық моделі	13
2 Қойылған мәселенің теориялық зерттеулері	15
2.1 Жүкті бөлшектеу шамасын анықтау және цикл құру	15
2.2 Ең кіші элемент тәсілі.....	15
2.3 Потенциал жүйесін құру	16
3 Деректер қорын құру	19
3.1 Деректер қорын жобалау.....	19
4 Тәжірибелік бөлім	21
4.1 Жобада таңдалған құрал жабдықтарды қарастыру	21
4.2 Серверлік PHP скрипттік тілінде программалау	21
4.3 Web 2.0 стандартының ерекшеліктері	22
5 Техникалық бөлім	23
6 Тіршілік қауіпсіздігі.....	28
6.1 ДК операторының еңбек жағдайларын сипаттау және жұмыс бөлмесіне талаптар	28
6.1.1 Кескіндеме және көрініс коэффициенттері.....	29
6.2 Ықтималды қауіпті және зиянды факторларды бағалау.....	29
6.3 Дербес компьютермен жұмыс кезіндегі қауіпсіздік талаптары.....	30
6.4 Жасанды жарықты есептеу	32
6.5 Табиғи жарықтандыруды талдамалы есептеу	35
7 Экономикалық бөлім	39
7.1 Жоба сипаттамасы және негіздемесі.....	39

7.2 Техничо-экономикалық тиімділігі	43
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР	45
ҚОСЫМША А.....	47

Кіріспе

Қазіргі уақытта математика мен қазіргі заманға есеп технологиясының жетістіктері экономикалық зерттеулер мен жоспарлауда кең қолданыс табады. Бұл математика, математикалық программалау сияқты, ойын теориясы, көпшілікке қызмет ету теориясы және де тез әрекеттегі электронды есептеуіш техникасының толқынды дамуы бөлімдерінің дамуына ықпал етеді. Математикалық программалаудың құрама бөлімдері сызықты, сызықты емес және динамикалық программалау болып табылады. Шешім шығару жолдардың жарату мүмкіндіктері болады, ол алгоритмдердің шешім шығару үшін электрондық есептеуіш машиналарын есептерге қолданады. Мұндағы басты рольді математикалық модельдеу атқарады. Математикалық модельді құру үшін зерттеліп жатқан жүйенің функционалдық мақсатын білу және шектеулер туралы мәліметтер қоры болуы қажет. Мақсаты және шектеулері функция түрінде ұсынылуы қажет. Осы дипломдық жобадағы жұмыс арқылы біз өзіміздің «Потенциалдар әдісі» пәнінен алған білімімізді бекітуге өз көмегің көрсетеді.

Модельдеу – бұл басқару жүйелерін синтездеу және талдау үшін қолданылатын ғылымның ең басты ерекше әдісі. Бұл ерекше тәсіл ұқсастықты тану орнына зерттейтін объектінің субъектісін таңдайды немесе оған қосалқы объекті бейнесін немесе үлгісін жасайды, оны зерттейді және жаңадан алынған білімді түпнұсқаға жинақтайды.

Оптимизациялық есептерде математикалық әдістерді пайдалану үшін, ең алдымен, тиімді шешімін табу қажет және есептің өзінің математикалық қойылымын жазуымыз қажет. Математикалық модель құру үшін, ең алдымен, зерттейтін объектінің ең басты қасиеттерін немесе заңдылықтарын біліп алып, оларды математикалық өрнектер арқылы өзара байланыстырып сипаттайды. Оптимизациялау теориясында мүмкін болатын шешімдердің ішінен мақсат функциясы деп аталатын функцияға ең үлкен немесе ең кіші мән беретін «ең жақсы» шешімді табу әдістерін зерттеу мәселелері қарастырылады. Мысалы, кез келген өндіріс фирмасының негізгі мақсаты – көбірек пайда табу. Жұмыс мақсаты: транспорттық есептерді шешуде өзінің мүмкіндіктерін тиімді пайдалану арқылы жоғары нәтижеге қол жеткізу. Есепті түрлі тиімдеу әдістерін пайдалана отырып шығарып, олардан алынған нәтижелерді салыстырып, нақты және тиімді шешім алу.

Жұмыс өзектілігі: математикалық әдіс арқылы экономикалық есептерді шешу және орнатуда едәуір нәтиже жинадық. Әсіресе оптимальды жоспарлау әдісі табысты дамуда, бұлар математикалық программалаудың мәні болып табылады. Негізгі болып оптимальді жоспарлаудың бірыңғай жүйесін құру есебі және халық шаруашылығын басқаруды математикалық әдіспен және электронды есептеуіш технологиясын экономикада қолдану.

1 Жоба тақырыбы бойынша аналитикалық талдау

1.1 Тасымалдауды ұйымдастыру негіздері

Көліктік үрдіс жылжымалы құрам беруге қатысты барлық операцияларды өзіне біріктіреді. Атап өтетін болсақ: жүк ағындарының маркетингі. Автомобильмен тасымалдаудың оң жақтары бар, Олар:

- тапсырыс беруші жүкті үнемі тікелей жүкті жіберушіден оны қабылдап алушыға дейін қайта артып-түсірусіз жеткізуге мүмкіндік береді;
- жүктің сақталу деңгейі жоғары, оған алаңдаудың қажеті жоқ;
- көлемі кішірек жүктерді де тасымалдауға қол жетеді;
- жүктерді теміржолға да, өзен және теңіз порттарына да қарамастан еркін тасымалдауға болады.

Ұзаққа жөнелту үшін автомобиль тасымалы жылдам бұзылатын, ерекше, бағалы, тез тасымалдау қажет, басқа жүк таситын көліктермен тасымалдауға ыңғайсыз жүктерді тасымалдауға пайдаланылады. Автомобиль тасымалының маңызы – қолайлылығы мен икемділігі, кемшілігі – тасымалдау құнының жоғары болуы.

Жүктерді жеткізудің ең көп тараған түрі автотасымалдау екені баршамызға мәлім. Басқа да жүк тасымалдау түрлері сияқты жүк тасымалының бұл түрінің де қолайлы және қолайсыз жағдайлары бар

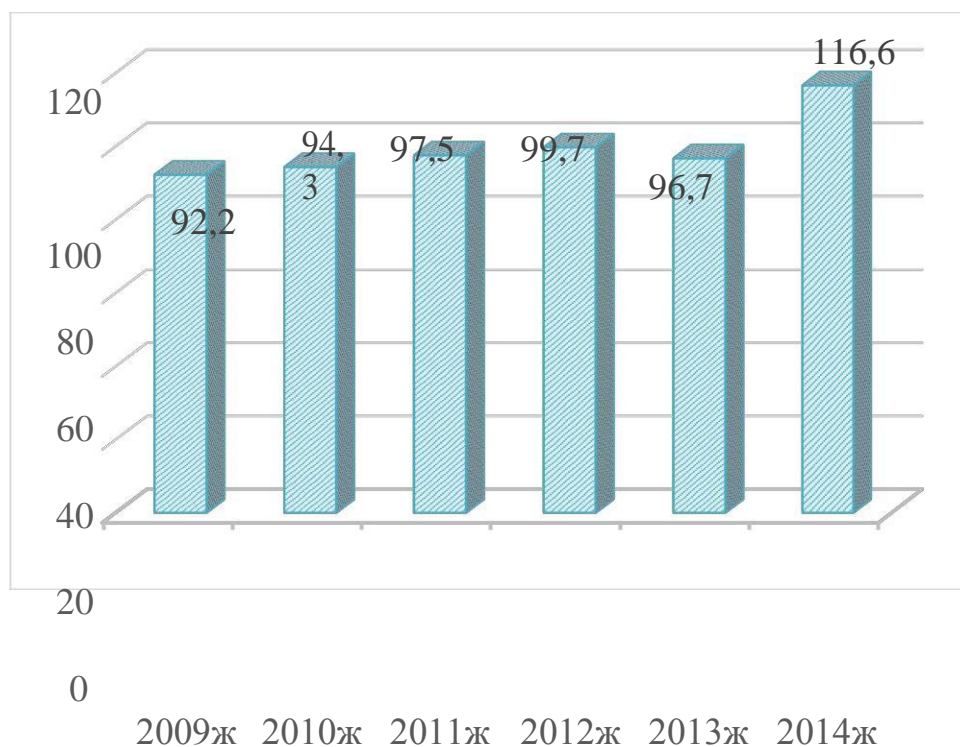
Қолайлылығына жүктерді теміржолмен тасыған кездегі (бұл әсіресе алыс қашықтыққа тасыған кезде білінеді) аралық жолдардың жетіспеуі есебінен аса жоғары ұтқырлық пен жылдамдықты атап өтуге болады.

Сонымен қатар, жүктерді тасымалдау ырғақтылығы болып табылады (яғни, жүкті көптеп жинақтаудың маңызы жоқ), жүкті міндетті түрде сақтауды қажет ету, жүктерді оның жіберушіден алушыға жеткенше қайта тиіп-түсірмей жеткізуге болады, жүкті жіберушілер алушылардың орналасқан жерінің теңіз және өзен порттарынан, соныменбірге ірі қалалардан, теміржолдардан тәуелсіздігі. Бұл қысқа жолға жүк тасымалдаудың ең қолайлы түрі болып табылады.

Қолайсыз жақтарының арасында өте ауыр жүктерді тасыған кезде алыс аймақтарғатасымалдауға мүмкіндіктің болмауын, жол бойы желілеріне тәуелділікті, көлемі аз жүктерді тасымалдаудың мүмкіндіктін жетіспеулерін бөліп көрсетуге болады.

Ұзақ қашықтыққа жүк тасымалдаған барысында көліктердің ішіндегі автотасымалдау түрі ең жоғарғы бағаға ие болады. Сонымен қатар қысқы кезеңде тасымалдау географиясы қысқарады.

Сурет 1.1 байқап отырғандарыңыздай, Алматы облысында автотасымалдау саны жыл артқан сайын жоғарлауды. Егер, 2013 жылы автомобиль көлігімен тасымалданған жүк саны 96,7 млн. тоннаға тең болса, 2014 жылы осы көрсеткіш 19,9 млн. тоннаға (немесе 20,6%) жоғарлаған және 116,6 млн. тоннаны құрап отыр.



Сурет 1.1 – Алматы облысында автотасмалдаудаға жүк, млн. тонна

Жылжымалы құрамды пайдаланудың қолайлылығын арттыру есептері және тасымалдаудың шығынын төмендету экономикалық – математикалық әдістерді пайдалану. Жүргізуші - көліктік тетіктерін орындауға қатысатын басты қатысушы, сондықтан да оның жұмысын қадағалау еңбек өнімділігінің деңгейіне тікелей қатысты. Тасымалдау және қауіпсіздік барысында жүргізуші жұмысы маңызды және үздіксіз дене жұмысымен тығыз байланысты.

Жүктерді тасымалдауды ұйымдастыру тасымалдау үрдісімен байланысып жатыр және тасымалдаушының техникалық мүмкіндігін мұқит пайдалануға мүмкіндік береді. Жылжымалы бөліктің түгел дерлік түрлерін тиімді пайдалану, олардың сыйымдылығы, жүктерді бағыттармен тасымалдауды көтеру, қозғалмалы құрамның бөліктердің бос тасмалданцын төмендету, жүк түсіру мен жүк тиеу кезінде автокөліктік транспорттың тежеліп қалу уақытын кеміту.

Тасымалдаушыға жүк тасымалдау жоспарға сәйкес мәлімдемелер жүк жөнелтушімен ГҮ-12 формасындағы бланкілерінде ресімделген жүк номенклатурасы анықталған үш данада тапсырылады. Өтінімнің әр данасына ұйым басшысы қол қойып, қол қаржылық операцияларда қолданылатын мөрмен бекітеді. Тасымал құрылымына сәйкес өтінімге енген мәліметтер және сандарының дұрыс жазылғанына, жүкті шығаруда ұйжым тұтынуы тасымалы мөлшеріне мәлімдемеге сай болуына, сондай-ақ, қозғалмалы құрам түріне, тасымалдың үнемділігіне жүк жөнелтуші жауапты болады.

Жүк тасымалдау шарттарына сйкес, айлық өтінімнің бланк формасы, электрондық есептеу машиналарында көрсетілген мәлімдемелердің

автоматтандырылған өңдеуінің мүмкіндігін пайдаланып, республикалық қатынастарда, халықаралық қатынастарда, үшінші елдің халықаралық қатынасында жүк жөнелтушілердің барлығында бірдей болады. 2010 жылы Республиканың жүктасмалындағы Алматы Облысының үлесі 5,9% құрап отыр.

Жолдың ардайым өзгеріске ұшырап отыратын жағдайы жүргізушіден үлкен мән беру мен тез нақты шешім қабылдауды талап етеді, сондықтан жүргізуші еңбегін құрастыру, құрастырылған еңбек режимі нормасын сақтауы қажет және жүргізушінің жұмыстан жалығуына жол бермеу қажет. Көлік үрдісін дұрыс қолпйлы етіп ұйымдастыру үшін мынадай талаптар ұсынылады: жүктерді тиеу және түсіруде автомобильдердің бос уақытының жоғары нормативті уақыт шығынын азайту; тиеу-түсіру жұмыстарын ауқымды теу және оларға комплексті механизация пайдалану; автомобильдер жұммысы кестесін жауаптылықпен құру және ұстану; автомобильдер маневріне аймақтар мен жол өткелдерін құрастыру, көп жағдайда тіркемелі автомобильдерге, тіркемелі немесе жартылай тіркемелі тягачтарға; апта бұрын жүктерді әзірлеу және т.б.

1.2 Жүк және жолаушы ағындарының сипаттамалары

Жүк автокөлік жұмыс барысы екі басты категориялармен түсіндіріледі: жүктерді жөнелту көлемі және жүк айналымы. Жүк ағыны нақты бір уақытта автомобиль көлігі тасымалдауында болмаса тасымалданғандағы тоннамен өлшенетін жүк көлемін анықтайды. Жүк айналым тонна-километрмен болмаса қазіргі уақытта немесе қандайда бір уақыт аралығында орындалған аралас жүктердің көліктік жұмыс жасауынан анықталады. Жалпы жүк айналым көлемі және автокөлік ұйымдарында жүк айналым құрастырылған наменклатураға сәйкес ажыратылады. Бұл ажырау жүк айналымы мен тасымал құрылысы қарастырылады. Тасымал көлемін анықтау барысында бір жүкті бірнеше рет тасуға болады (бұл көбіне қалалы орындарды кездеседі). Бұл дегеніміз бір жүкті қабылдау орнына жеткізудегі айырбастаушылық болуы мүмкін. Мысал кетіретін болсақ, кейбір жабдықтауға арналған құрылыс керек-жарақтары, санитарлы-техникалық және электромонтаждау жұмыстарында басты, сәйкес қоймаларға жеткізіліп, содан кейін құрылыс нысандарына жөнелтіледі. Көбінесе қайталанбалы тасымалдар сауда жүйелерінде индустриялық және азық-түлік тауарлары болып табылды. Көбінесе бұл тауарлар әуел баста темір жолмен, су және әуе көліктермен станция немесе аэропорттардағы сауда қоймаларына жеткізіліп, тазартылып, сосын дүкендерге жөнелтіледі.

Уақыттың жеткіліктігіне байланысты жүк айналым және тасымал көлемі сағаттық, тәуліктік, айлық, кварталдық, жылдық болып ажыратылады. Жүк айналым және тасымал көлемі жоспарға сәйкес әрбір ай және квартал бойынша бір сарында таратылмайды. Бұндай құбылыс автокөлік жұмысы мен сұранысқа қатысты болады. Бір қалыпсыздық деңгейін бір қалыпсыздық коэффициентпен анықтауға болады, тасымал шамасы жүк айналымының максималды көлемімен қандайда бір уақыттың орта мәнді қатынасына сәйкес келеді.

1.3 Жүк ағындарының сипаттамасы

Жүк ағындары белгілі бір уақыттағы белгілі-бір бағыт бойынша тонналық жүк көлемдері. Сызба түрінде немесе жүк эпюрасы түрінде ағындар графикалық жүк ағындары болып табылады. Қозғалмалы құрам тасымалдауындағы қатынас жолдарында жүктің қисық сызықты барысы түзу сызықтықпен ауыстырылады. Жүк ағындарының эпюрасы мынадай түрде беріледі. Бастысы, белгілі бір ұзындық масштабымен тасымалдау кезіндегі бір немесе бірнеше аудандар анықталады. Содан соң осы сызыққа белгілі бір масштабпен тасымал қашықтығын біріктіріп алғандағы жүк саны бейнеленеді: бірінші кезекте жүк, келесі де қабылдау пункті болып табылады. Жүк тасымалының эпюрі түзу және кері бағытта сызыллып бейнеленеді. Түзу және кері бағыттағы жүк ағындары қатынасының саны бағыт бойынша жүк ағындарының бірқалыпсыздық коэффициенті деген атқа ие болады. Жүк ағындары эпюрасын тұрғызуда мынадай мүмкіндіктері болуы мүмкін: барлық жүк жіберетін және жүк қабылдайтын орындар.

Жүк ағындары картограмма түрінде беріліп жатады. Картограмма әртүрлі жүктердің арнайы байланыстағы автомобиль жолдары, көшелердегі жүк ағындарының картадағы графикалық берілуі. Эпюр мен картограмма жүктасымалындағы жылжымалы құрамның маршруттық қозғалысын ашып көрсету үлкен сапаға ие болу мүмкіндігін береді.

1.4 Жүкті тасымалдауға қабылдау.

Жүктер көбісі өздеріне тән қасиеттерге байланысты қабылдау, тасымалдау барысында берілген шарттарды бағынуды және оларды қатаң сақай білу қажет. Бұл жүктер қатарына: қауіпті, тезбұзылатын, қатып қалатын, габаритті емес және экспорттық жүктер жатқыза аламыз.

Жүктерді көбнесе әртүрлі қораптарда (тара) ыдыстарда тасымалдауға болады. Көп кездесетін ыдыс түрлері: қатты (жәшіктер, бөшекелер, бидондар), жартылай қатты (қорап, шарбақ, корзина) және жұмсақ (мешок, сетка).

Ыдыстың көптеген түрлерін қайтара пайдалануға болады. Кез келген тауарды салуға пайдаланатын әмбебап ыдыстар болады және белгілі бір тауар түрлерін салуға ғана арналған (цементке арналған мешоктар, кислоталарға арналған бөтелкелер) арнайы ыдыстар болып ажырытылады.

Көптеген жүктерді қораптарға орналастыру амалдары арнайы белгіленген техникалық шарттармен жүзеге асырылады, орамасында стандартта берілмеген жүктер де бар. Көп жағдайда да ыдыс тасымалдауда жүктің сақталуын мерзімін анық көрсетуі қажет.

Тасымалданатын жүктің сақталу мерзімін және оның уақытысында тиісті орынға жетуін қамтамасыз етудегі бастысы, оның жөнелтілген пункті және жолы, бару станциясы және жолы, қабылдаушының аты-жөні, жіберушінің белгілері және маркалары көрсетілетін көліктік маркировкасын болып табылады.

Жүктің жеке түрлеріне жіберуші, оны тиеу- түсіру жұмыстары барысында және сақтауда қасиеттерін көрсететін арнайы манипуляциялық белгілерді анық көрсету қажет. Мысал келтіретін болсақ: «Байқаңыз», «Не кантовать», «Жоғары» және т.б.

Қоғамдық пайдалану орындарында (станциялық қоймаларда және алаңдарда) жүкті тасымалдауға қабылдап алу, жүкті станцияға жеткізуге рұқсаты қойылған алдын ала толтырылған құжаттарға(накладной) сәйкес жүзеге асырылады. Әкелу рұқсатын тбақылай отырып, қабылдап тапсырушы жүкті жіберушінің көрсеткен жүгі, қамтылған орындар саны және сыртқы орамының тегі құжатта берілгендермен сәйкес келуін қадағалау қажет. Сыртына қарап тексеруде қорапта немесе орымында, жүктің сақталуын қамтамасыз жасай алмайтын, жеткіліксіздіктер байқалса, онда қабылдап тапсырушы ол жеткіліксіздіктердің жойлуын талап етуі қажет болмаса тасымалдауға кемеитін (нетранспортабельный) жүкті тасымалдауға қабылдап алдан бас тартқаны абзал.

Қораптардағы және бөлектенген(штучных) жүктерді қабылдап алуда, жүк жіберушінің құжатта олардың салмағы және оны анықтау амалдарының бейнеленуін, қабылдап тапсырушы тексеріп алуға міндетті болып табылады. Жүктің көлемі көлем өлшеуіштерде өлшеп немесе трафаретке сәйкес немесе стандарттарға сәйкес жүк орындарындағы салмағын есептеу арқылы анықтауға болады. Егер жүктің салмағы трафаретке сәйкес анықталған болса, онда оның дұрыс анықталғанына және мұндай трафареттердің жүк орындарында бар екендігіне анықтау қажет[20].

Жүкті және оның орамын анықтап тексерумен бірге оның көліктік маркировкасын анықтау қажет. Тасымалдауға қабылданған жүкті қабылдап алып өткізуші жүкті жіберуге қабылдау кітабына енгізуі қажет(форма ГY-34).

1.5 Желілік программалаудың транспорттық тапсырмасының берілуі және математикалық моделі

Желілік программалаудың (ЖП) көптеген тапсырмаларының арасынан арнайы тапсырмалардың қандайда бір топтары бөлініп алынады. Оларды барлық әдістерімен шешу есептік сипаттағы бірқатар қиындықтармен аралас келдеі. Осыған байланысты белгілі, едәуір қарапайым әдістер шешу үшін ұсынылады. Тапсырмалардың осындай топтарының бірі ретінде транспорттық тапсырмаларды атап өтсек болады[21].

Тапсырманың ұсынылуы. A_1, A_2, \dots, A_m -нің m жіберілу пункттерінде біртекті жүктің, сәйкес a_1, a_2, \dots, a_m бірлігі бар, ол B_1, B_2, \dots, B_n -нің n тұтыну пункттеріне апару қажет (1.1 – кестеде көрсетілген). Ал оларға сұраныс сәйкесінше v_1, v_2, \dots, v_n бірлігін құрап отырады. Мұнда A пунктінен B пунктіне ($i=1, m, j=1, n$) жүк бірлігін тасымалдауға жұмсалатын транспорттық шығындар C беріледі. Барлық жүк жіберілу пунктінен толығымен шығарылатын, барлық тұтыну пункттеріндегі сұраныс қамтамасыздандыратын және транспорттық шығындар сомасы ең төмен болатын тасымалдау жоспарын анықтап алу қажет, тапсырманың алғашқы мәліметтер бөлуші 1.1 кестеде берілген.

Кесте 1.1 – Тапсырманың алғашқы мәліметтерін бөлуші

b_j	b_1	b_2	...	b_j	...	b_n
a_i	c_{11} x_{11}	c_{12} x_{12}	...	c_{1j} x_{1j}	...	c_{1n} x_{1n}
a_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2j} x_{2j}	...	c_{2n} x_{2n}
...
a_i	c_{i1} x_{i1}	c_{i2} x_{i2}	...	c_{ij} x_{ij}	...	c_{in} x_{in}
...

Тапсырманың математикалық моделін құрастыру үшін үшін A_i ($i=1, m$) пунктiнен B_j ($j=1, m$) пунктiне тасымалданатын жүк бірлігiнiң көлемiн көрсететiн $X_{ij} \geq 0$ ($i=1, m, j=1, n$) айнымалыларын ұсынымыз.

2 Қойылған мәселенің теориялық зерттеулері

2.1 Жүкті бөлшектеу шамасын анықтау және цикл құру

Тасмалдауға байланысты бірлік жүгінің мәнін есептеу үшін бос торды “+” таңбасымен белгілеп, жүктеу керек. Бұл толтырылған ұяшықпен бос ұяшықтың байланысын көрсетеді. Кестеде толтырылған торларда $m+n$ болып табылады. Сондықтан цикл түзіледі, ”+” белгіленген ұяшықты шығару қажет және ол цикл бір ғана болып табылады. Циклды анықтап “+” таңбасымен таңбаланған ұяшықтың қозғалысын бастап, “-“ және “+” таңбаларын кезектестіріп орналастырып отырамыз. Содан соң $0_0 = \min x_{ij}$ анықтап, мұндағы x_{ij} - тасмалдаушы, ол циклдың басында орнын алады және де “-“ таңбасымен таңбаланады. 0_0 шамасы анықталған шамада жүктің неше бірлігін бөлшектеу қажет болатынын анықтап аламыз. 0_0 мағынасын “+” таңбасымен таңбалап бос ұяшыққа тіркеп, циклмен жылжи отырып 0_0 -ді оқып ұяшыққа орналасқан көшіру көлемін “-“ таңбасымен таңбалаймыз және көшіру көлеміне біріктіріп қосып аламыз. Ұяшықта орналасқанын “+” таңбасымен таңбалаймыз. Егер кішірейтілген көшулер 0_0 бойынша келетін болса, онда есептеу барысында тура келетін ұяшықты нөлдік тасмалдаумен тірек жоспарында толтырылған ұяшықтар $m+n-1$ мәнге тең болуы қажет [27].

Анықтап отырған A_2B_4 ұяшығын “+” таңбасымен белгілеп және 1.1 – кестеде берілген циклды анықтаймыз. $0_0 = \min (50; 50; 0) = 0$ т.с.с нөлдік тасмалдауды A_2B_4 ұяшығына орналастыру керек, қалған сандарды анықтау мен нөлге біріктіру анық және өзгеріске келмейді.

Бөлшектеуді қорға келгенде 0_0 жаңа анық емес тірек жоспары құрылып шығады, оның тиімділікке байланысы бар. Жаңа тірек жоспарының пайдалы жағын анықтау үшін тағы потенциал жүйесін құру керек және кез-келген ұяшықтың пайдалы шартының орындалуын анықтап алуға болады.

Егер есептелген жоспар тағы тиімді болмаса, онда қайта есептеуді қажет етеді. Бос ұяшықтар қанағаттандырылғанша бұл есептеулер қайталана жүргізіледі. Тиімділік жоспарының барлық есептеулері 1 ұяшықта жүзеге асу ұсынылады. Сондықтан x_{ij} ; U_i ; V_j және V , ”+”, ”-“ таңбаларын карандашпен жазып алған дұрыс, себебі кейін өшіріп тастауға мүмкіндік туады.

2.2 Ең кіші элемент тәсілі

Жоғарыда қарастырып өткен әдіс бойынша үлестіру барысында тасымалдауға жұмсалатын қаржыны сепке алуға да болмайды. Ол үшін ең төменгі элемент тәсілін қолдансақта болады. Ең төменгі элемент тәсіліне сәйкес кестені ең кіші элемент орналасқан ұяшықты толтырудан бастауға болады (кесте 2.1). Қандайда бір элемент тәсілін алып қарастырсақ.

Кесте 2.1 – Ең кіші элемент тәсілі

	Тұтынушылар	Жүк
--	-------------	-----

қоймалар	B1	B2	B3	B4	B5	Қоры
A1	8 0	2 110	8 0	3 40	6 0	150
A2	2 100	8 0	4 10	7 0	6 130	240
A3	4 0	3 0	2 80	4 60	8 0	140
қажеттілік	100	110	90	100	130	530

Бұл есептегі ең кіші элемент 2 мәніне ие; ол үш рет кездеседі; олардың кез-келгенінен бастасақта болады[15]. Біз бірінші жолда орналасқанынан анықтай бастасақ. Кестенің осы кереге көзіне 110 санын белгілеп аламыз, немесе екінші тұтынушы B_2 өзіне қажетті барлық жүкті бірінші қоймадан ең төменгі бағамен алады, немесе $x_{12}=110$; ол өзге қоймадан жүк алмайды: $x_{21} = x_{32} = 0$

Бірінші қоймадан қалған 40 мөлшердегі жүк төртінші тұтынушыға тиісті болады, себебі оның беретін бағасы басқаларына қарағанда төмен. Осылайша кестенің барлық ұяшықтарын толтырып шығуға болады. Сонымен қатар жаңа жоспар:

$$\begin{aligned} x_{11} &= 0; x_{12} = 110; x_{13} = 0; x_{14} = 40; & x_{15} &= 0; \\ x_{21} &= 0; x_{22} = 0; x_{23} = 10; x_{24} = 0; & x_{25} &= 130; \\ x_{31} &= 0; x_{32} = 0; x_{33} = 0; x_{34} = 60; & x_{35} &= 0; \end{aligned}$$

Бұл жоспарды жүзеге асыруға жұмсалатын қаржы көлемі:

$$F = 2 \cdot 110 + 3 \cdot 40 + 2 \cdot 100 + 4 \cdot 10 + 2 \cdot 80 + 4 \cdot 60 + 6 \cdot 130 = 1760$$

Осы әдіске сәйкес анықталған жоспар диагональды әдісінен пайдалырақ екендігін көруге болады.

Ескерту. Тасымалдау кестесін толықтыру барысында толтырылған ($x_{ij} = 0$) ұяшықтар саны $n+m-1$ –ге тең болуы қажет.

2.3 Потенциал жүйесін құру

Потенциал жүйесін құрастыру барысында $U_i + V_j = C_{ij}$ шарты қолданылады. Мұнда C_{ij} -j бағаны және i жолының толық ұяшығындағы жүк бірлігінің бағасы.

Потенциалдар жүйесінің анық емес тірек жоспары үшін ғана құрастыруға болады. Мұндай жоспар $m+n-1$ толық ұяшығын құрап отыр. Сондықтан оған $m+n-1$ сызықтық тәуелсіз түрін түзетумен қатар $n+m$ тәуелсіз жүйелерін құруға мүмкіндік бар. Белгісіздерден түзетулер 1 қадам кіші болады, сондықтан жүйе белгісіз болады және 1 белгісізге (әдетте U_i) 0-дік мағынаға ие. Содан соң қалған потенциалдар бірегей анықталып отырады. U_i потенциалы табылған

болсын, одан $V_j = C_{ij} - U_i$ анықталады [15,18,27].

Егер V_j потенциалы анықталған болса, онда одан $U_i = C_{ij} - V_j$ -ді алып тастаймыз. Мұндай әдіспен беймәлім потенциалды табу үшін C_{ij} шамасындағы толық ұяшықтағы белгілі потенциалдарды тауып алуға тура келеді.

1 бос емес ұяшық болмағаны үшін, тірек жоспары анық көрсетіледі. Сондықтан толық ұяшықтардың санын көбірек аламыз (A_4 жолы), және $U_4 = 0$. A_4 жолында 3 толық ұяшық орналасқан (A_4B_2, A_4B_3, A_4B_5), олар U_4 потенциалы бойынша V_2, V_3, V_5 потенциалдарымен бірге. $V_2 = C_{42} - U_4 = 8 - 0 = 8$, $V_3 = C_{43} - U_4 + 12 - 0 = 12$, $V_5 = C_{45} - U_4 = 13 - 0 = 13$.

Осы потенциалдарды анықтап алайық. U_4 потенциалын пайдалана отырып кез-келген потенциалды табуға болмайды, соған байланысты оны \square таңбасымен белгілеп аламыз. Енді ретімен потенциалдары анықталған B_2, B_3 және B_5 бағандарын тексеріп шығамыз. B_2 бағанында 2 толық ұяшық орналастырылған (A_2B_2 және A_4B_2), ол V_2 потенциалын U_2 және U_1 потенциалын U_2 және U_1 потенциалымен байланыс жасайлады, U_4 потенциалы табылады. C_{22} көмегімен $A_2 B_2$ ұяшығына көшіп белгісіз потенциалды анықтап аламыз. $U_2 = C_{22} - V_2 = 7 - 8 = -1$ V_2 потенциалын \square белгісімен белгілеп, B_3 бағанына көшіріп аламыз. Бұл бағанда толтырылған V_3 потенциал белгісіз потенциал бағандарымен байланыстыратын толтырылған тор көз жоқ. V_3 потенциалын \square белгісімен белгілеп, B_5 бағанына көшіріп аламыз. Мұнда 1 толық (A_3B_5) тор көздері бар, ол V_5 -ті белгісіз U_3 потенциалымен байланыстырады. Оны анықтап аламыз: $U_3 = C_{32} - V_5 = 2 - 13 = -11$. V_3 потенциалын \square белгісімен белгілейміз, белгісіз потенциалдар бағаны қолдануымызға байланысты, белгілі потенциалдар бағанына көшіреміз. Олар әлі \square белгісімен белгіленген, содан кейін оларға сәйкес келетін бағандарды тексереміз. U_2 потенциалы $A_2 B_1$ толық тор көздерімен, V_1 белгісіз потенциалымен байланыс жасалған. Мына потенциалдарды қарастыратын болсақ: $U_1 = 2 - (-1) = 3$ және

U_2 потенциалын белгісімен таңбалап аламыз. A_3 тізбегінде U_3 потенциалын беймәлім потенциалдармен байланыстырып толтырылған тор көздер жоқ. U_3 потенциалын \square белгісімен таңбалаймыз. \square белгісімен таңбаланған белгілі потенциалдар бағанына көшіреміз (V_1 потенциалы). Бірақ B_1 бағанында V_1 - байланыстыратын беймәлім потенциалдар жолымен толықтырылған тор көздер жоқ. Сонған орай V_1 потенциалын \square таңбасымен белгілеп аламыз. Потенциалдар жүйесінің құрылымы үзіледі өйткені, U_1 және V_4 потенциалы анықтаусыз қалады. Бұл тірек жоспарының анық емес болуына байланысты болады (1 толтырылған ұяшықтың жоқтығынан). Анықтылықты орнықтыру үшін толтырылған ұяшықтың мәнін 0-дік тасымалдауды еңгізу арқылы $m+n-1$ -ге дейін толтырады. 0-дік тасымалдаулар еңгізілген тор көздер фиктілік толтырылған деген атқа ие болады.

U_1 және V_4 потенциалын табу барысында толтырылған ұяшықтың біреуін фиктілеу қажет, A_1 жолын немесе B_4 бағанының 1 потенциалы анықталған болуы қажет. Тапсырма сызықтық минимизация функциясымен шешіміг табады. Сондықтан бағасы ең кіші ұяшықтар толтырылған деп аталады.

Бағасын тексеріп, толтырылған тор көздердің A_1 жолы және B_4 бағанындағы ең кішісін анықтап ($\min C_{ij} = 2$), A_3B_4 ұяшығына сәйкес болатындай етіп, оған 0-санын жазып оны толық ұяшық ретінде қарастыруға болады. Енді $A_3 B_4$ ұяшығы V_4 потенциалымен U_3 потенциалын байланыс жасаймыз; $V_4 = C_{34} - U_3 = 2 - (-11) = 13$. Одан кейін $U_1 = C_{14} - V_4 = 1 - 13 = -12$ анықтаймыз.

Потенциал жүйесі, яғни потенциал бейнелетін □ белгісімен жоюға тура келеді. Жүйенің құрылу талаптарын тексеріп аламыз. Оған толықтарған ұяшықтардың жолын тексеріп және потенциалдың әрбірінің бағасын анықтап аламыз. Егер барлық толықтырылған ұяшықтардың теңдігі орындалып жатса, онда жүйе дұрыс құрылған болып есептелді. Кері жағдайда оны қайта құрып немесе шарт орындалатындай етіп өзгертуге тура келеді.

3 Деректер қорын құру

3.1 Деректер қорын жобалау

Жобада деректер қоры жүйесі ретінде MySQL қолданылды. Дерекқорларын құруға, өзгертуге және басқаруға SQL тілі қолданылды. Деректер қорындағы негізгі таблицалары:

- Goods – тауарлар
- Users - қолданушылар

- Roads – жолдар
- Stations – станциялар
- Etsng – тауар типі
- Wagons – вагондар
- Wagon_types – вагондар типі
- Countries – мемлекеттер

Goods таблицасында тасымалданатын тауар турады ақпарат сақталады. Мұнда тауардың шамалары, типі және тағы басға ақпарат сақталады:

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1 id	int(10)		UNSIGNED	Нет	Нет
2 id_crm	varchar(255)	utf8_unicode_ci		Нет	Нет
3 date_create	timestamp			Нет	0000-00-00 00:00:00
4 etsng	int(10)		UNSIGNED	Нет	Нет
5 wagon_type	int(10)		UNSIGNED	Нет	Нет
6 weight	double(8,2)			Нет	Нет
7 wagon_size	double(8,2)			Нет	Нет
8 from_station	int(10)		UNSIGNED	Нет	Нет
9 to_station	int(10)		UNSIGNED	Нет	Нет
0 price	double(15,2)			Нет	Нет
1 currency	varchar(255)	utf8_unicode_ci		Нет	Нет
2 tariff	tinyint(1)			Нет	Нет
3 from_date	timestamp			Нет	0000-00-00 00:00:00
4 to_date	timestamp			Нет	0000-00-00 00:00:00
5 shipping_count	int(11)			Нет	Нет

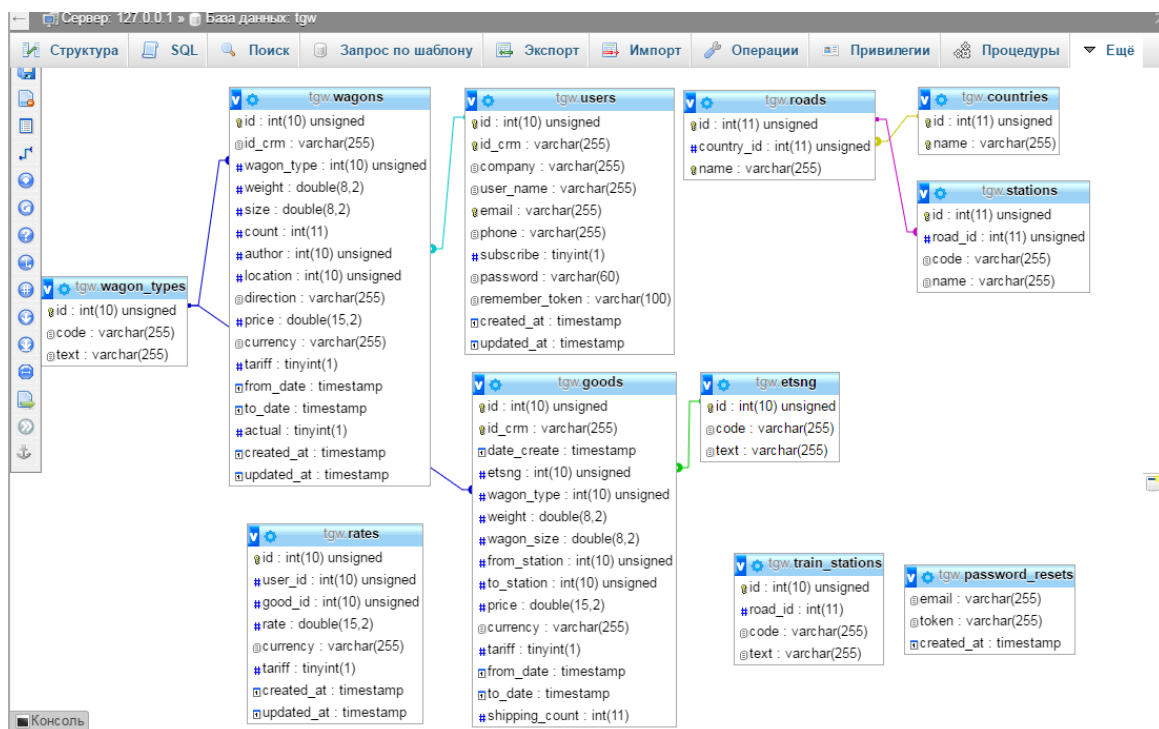
3.1.1 – сурет. Goods таблицасы

Wagons таблицасында вагондар туралы негізгі ақпараттар сақталады. Мұнда вагондардың сымдылығы, орналасқан жері және басқа да ақпараттар сақталады:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	id	int(10)		UNSIGNED	Нет	Нет
2	id_crm	varchar(255)	utf8_unicode_ci		Нет	Нет
3	wagon_type	int(10)		UNSIGNED	Нет	Нет
4	weight	double(8,2)			Нет	Нет
5	size	double(8,2)			Нет	Нет
6	count	int(11)			Нет	Нет
7	author	int(10)		UNSIGNED	Нет	Нет
8	location	int(10)		UNSIGNED	Нет	Нет
9	direction	varchar(255)	utf8_unicode_ci		Нет	Нет
10	price	double(15,2)			Нет	Нет
11	currency	varchar(255)	utf8_unicode_ci		Нет	Нет
12	tariff	tinyint(1)			Нет	Нет
13	from_date	timestamp			Нет	0000-00-00 00:00:00
14	to_date	timestamp			Нет	0000-00-00 00:00:00
15	actual	tinyint(1)			Нет	1
16	created_at	timestamp			Нет	0000-00-00 00:00:00
17	updated_at	timestamp			Нет	0000-00-00 00:00:00

3.1.2 – сурет. Wagons таблицасы

Деректер қорында басқада маңызды кестелер бар. Олар бір-бірімен белгілі бір тәсілмен байланысқан. Бұл байланысу тәсілдері деректер қорымен жұмыс жасауға, деректер енгізуге және өңдеуге оңтайлы тәсілмен құрылған.



3.1.3 – сурет. Деректер қорының сұлбасы.

4 Тәжірибелік бөлім

4.1 Жобада таңдалған құрал жабдықтарды қарастыру

Жобаны іске асыруда ашық қайнарлы бағдарламалар қолданылды. Мұндай таңдаудың бірнеше артықшылықтары бар:

- Ашық қайнарлы бағдарламалар тегін болғандықтан жобаның өзіндік құны төмен.
- Ашық қайнарлы бағдарламаны белгілі бір компания емес, тәуелсіз бағдарламашылар бірлестігі іске асырады және техникалық қолдау жасайды.
- Мұндай бағдарламалық қолжетімді және кең таралған.

Операциялы жүйе ретінде Debian GNU/Linux негізіндегі Ubuntu 14 қолданылды. Ubuntu веб-сервердің басым көпшілігінде қолданылады. HTTP сервер ретінде Nginx қолданылды. Nginx жоғары жүктелген жүйелерде кеңінен қолданылады. Өзінің ерекше архитектурасының арқасында бір мезетте миллиондаған сұраныстарды орындай алатын болғандықтан Nginx Яндекс, Mail.Ru, ВКонтакте және Рамблер секілді жоғары жүктелген жүйелерде қызмет етеді. HTTP-сервердің негізгі қызметтері:

- Статикалық сұраныстарды өңдеу, индекстік файлдарды құру, файлдар тізімін автоматты түрде жасау, ашық файл дескрипторын кәштеу
- Жүктемені бөлу мен жүйенің тоқтаусыз жұмысын қамтамасыз ету
- FastCGI , uwsgi , SCGI және Memcached серверлерін жеделдетілген қолдау
- Модульдік, сүзгілер, оның ішінде қысу(gzip), byte-ranges, SSI- сүзгі, XSLT- сүзгі
- SSL және TLS SNI қолдау
- HTTP/2 және салмағы мен тәуелділік негізінде басымдықтарын қолдау.

Жүйе жұмысын оңтайландыру үшін Memcached қолданылды. Memcached–ақпаратты жүйелік жадыға хэш-таблица негізінде кәштейді.

4.2 Серверлік PHP скрипттік тілінде программалау

Объекті бағдарланған программалау – қазіргі таңдағы программалау тілдерінің негізгі технологиясы болып табылады. Бұл технологияда басты мүше объект болып табылады және программаның өзге бөлігінен тәуелсіз, кейбір функционалдық есептерін анықтау барысында оның бөліктері де жеткілікті болып отырады. PHP - Веб-серверде HTML беттерін құру және дерекқормен жұмыс істеуге арналған скрипті бағдарламалау тілі. Қазіргі күндерде хостинг қызметін ұсынатындардың көпшілігімен қолданады. LAMP веб-торап жасауға арналған стандартты жинақталымына енеді. Тілдің оңайлығы, орындалу жылдамдығы, функциялық байлығы және PHP негізіндегі әуелгі кодтардың

кеңге таралуының арқасында PHP тілі Торда программалау саласында ең белгілі тілдердің бірі боп есептелінеді (JSP мен ASP-де қолданылатын тілдермен қатар). Бұл тілдің айрықшылығы ядро мен соған қосылатын модульдарында. Соңғылары дерекқор, сокет, динамикалық графика, криптографиялық кітапханалар, PDF форматты құжаттармен және т.б. жұмыс істеуге арналған. Бұндай модульды қалаған адам дамытып қоса алады. Модульдардың саны бір неше жүз болғанымен, стандарттық жабдықтауға тек жақсы нәтиже көрсеткен бір неше оны ғана кіреді. PHP интерпретаторы веб-серверге не арнайы сол сервер үшін жасалған модуль арқылы (мысалы, Apache әлде IIS), не CGI-қосымшасы ретінде қосылады.

Бұған қоса, UNIX, Linux, Windows және Mac OS X амалдық жүйелерінде әкімшілік тапсырыстарын атқаруға пайдаланылуы мүмкін. Бірақ PHP тілі бұл салада кең өріс алмаған, бұған қарағанда Perl, Python және VBScript тілдері айтарлықтай нәтиже көрсетіп отыр.

Тілдің синтаксисі Си тәріздес. Ассоциативтік массивтар мен foreach циклі секілді кейбір элементтері Perl тілінен алынған.

Қазіргі заманда PHP жүздеген мың дамытушылармен пайдалануда. Интернет желісі сайттарының 5-тен бірі осы тілде жазылған.

PHP дамытушылар тобы тілдің ядросы, қосымшаларымен жұмыс істейтін, соған қатар PEAR не тілдің құжаттамасы сияқты байланысты жобалармен айналысатын көптеген адамдардан тұрады.

4.3 Web 2.0 стандартының ерекшеліктері

Web 2.0 немесе Уеб 2.0 ғаламтордың (world wide web) интерактивтік және коллаборативтік, яғни істестік элементтерін білдіруге қолданылатын термин. Демек, қолданушы интернеттегі мазмұнды тұтынып қана қоймай, сонымен қатар өзі де қандай да бір мазмұн өндіреді. Бұл ұғым кейінгі кезде әлеуметтік желілердің дамуымен үлкен маңызға ие болып келе жатыр.

Қолданушылар интерактивті қосымшалардың көмегімен мазмұнды өздері жасап, өңдеп, оны таратады. Қолданушылардың жаңа ролін сипаттау үшін просьюмент (ағыл. "Prosumer") деген түсінік пайда болған. Контент, яғни мазмұн бұрынғыдай орталықтандырылған түрде медиа компаниялары тарапынан емес, саны біршама көп, өзара әлеуметтік желілер арқылы байланысқан қолданушылар тарапынан дайындалып және интернетте таралып келеді.

Контентті көруге ғана мүмкіндік беретін веб-сайттармен салыстырғанда, Web 2.0 ерекшелігі сол, ол қолданушыларға әлеуметтік медиа арқылы виртуалды қауымдастық құруға мүмкіндік береді, бір-бірімен байланыстырып береді және бірге істесіп жұмыс жасауға жағдай жасайды. Web 2.0 мысалдарына әлеуметтік желі сайттары, блогтар, wiki-лер, видео бөлісетін сайттар, веб-қолданғылар (application), mashup және folksonomy-лер жатады.

5 Техникалық бөлім

Жобаны іске асыруда жүйенің жұмыс жасау жылдамдығына көп көңіл бөлінді. Себебі, операторлар қажетті ақпаратты барынша тез алуы тиіс. Техникалық орындау кезінде қазіргі заманғы барынша өнімді технологиялар мен тәсілдер қолданылды. Мәселен, ақпаратты сүзгілеу, енгізу, өңдеу, т.б. үшін ажах технологиясы кеңінен қолданылды.

Жүйе логикасын, деректер қорымен жұмыс жасау қызметін, кәштеу мен сессиялармен жұмыс жасау және т.б. үшін Laravel Framework қолданылды. Laravel-дің басқа бәсекелестерінен артықшылығы сұранысты өңдеу жылдамдығында. Laravel өте тез жұмыс жасайды. Оның көмегімен бағдарлама іске асыру оңай әре тез. Бұл, өз кезегінде, жүйені жасауға қажет уақытты қысқартуға мүмкіндік береді. Laravel-дің бір кемшілігі: операциялық жадында көп орын алады. Сол себепті, жобаны фирма жұмысына енгізгенде операциялық жады көп сервер алынды. Қосымша жадыға кеткен шығын жұмыс жасау жылдамдығы мен жобаны іске асырудың жеделдігімен толық ақталды және алға қойған мақсатқа қол жеткізуге мүмкіндік берді. Яғни, біз сұранысты өңдеу жылдамдығы жоғары жүйені аз уақытта іске асырдық.

Жүйеге кіру үшін операторлар логин және құпия сөзді енгізуі тиіс. Егер бір қолданушы бірнеше рет қатарынан қате ақпарат енгізсе лог жазылып, администраторға хабарлама жіберіледі.

YTMPartner

Email

Пароль

Запомнить меня

Войти

[Забыли пароль?](#)

220 ВИДОВ ГРУЗА	1000 ПОТЕНЦИАЛЬ.. ПЕРЕВОЗОК	100 ТРЕБУЕМЫХ ВИДОВ ПС	500 ЗАПРОСОВ НА РАСЧЕТ
--------------------	-----------------------------------	------------------------------	------------------------------

5.1 сурет. Жүйеге кіру

Жүйеге кірген соң, оператор өзіне керек ақпаратты алады. Немесе жүйеге жаңа ақпарат енгізеді. Жаңа ақпарат енгізу үшін оператор арнайы форманы толтыруы тиіс. Жүйеге енгізуге болатын ақпараттың бірнеше түрі бар: тасымалдауды қажет ететін тауар және тауар тасымалдайтын көлік.

Тип вагона *

Выберите тип вагона ▾

Тоннаж т

Объем м³

Количество вагонов шт

Дислокация *

Станция ▾

Желаемое направление *

Страна ▾

Дорога ▾

Станция ▾

Ставка за пользование *

КЗТ ▾

Актуальность запроса с момента размещения информации

▾

5.2 сурет. Жүйеге жаңа ақпарат енгізу

Оператор өзіне қажет ақпаратты арнайы сүзгілердің көмегімен тез әрі оңай ала алады. Сүзгілер аях технологиясының көмегімен үлкен көлемдегі ақпаратты лезде өңдейді. Тасымалданатын тауарлар зүзгісінде тасымалдайтыг көлік түрі, арналасқан жері, тасымалдау жері, салмағы, тасымалдау уақыты сүзгілері бар. Тауарлар тізімінде тауар туралы негізгі ақпарат көрсетіледі. Тізімде бір тауарды таңдаған сәтте сол тауар туралы толық ақпарат көрсетіледі

Поиск грузов для перевозки

Фильтр

Тип вагона
Введите тип ваго

Станция погрузки
Введите станцию

Станция выгрузки
Введите станцию

Вес
от т до т

Дата перевозки
с по

Мои ставки
[Сбросить фильтр](#)

Сортировать по: [дате добавления](#) / [дате перевозки](#)

02.06	Бой стеклянный 69 т	полувагон	Железородная (6839... Алма-Ата I (700007...	—	Проходная ставка: Не указано Предложите ставку
01.06	Ангидрит (шпат полев... 69 т	полувагон	Чайкурук (706906) Алма-Ата I (700007...	—	Проходная ставка: Не указано Предложите ставку
30.05	Насосы, не поименова... 15 т	крытый	Алма-Ата I (700007... Актобе (667909)	—	Проходная ставка: Не указано Предложите ставку
30.05	Кирпич асбоцементный 68 т Отгрузок в месяц: 1	крытый Объем вагона: 138 м³	Аксенгер (рзд) (70... Тахияташ (758506)	—	Проходная ставка: Не указано Предложите ставку
30.05	Камень известняковый... 69 т	полувагон	Добрятино (243309) Алма-Ата I (700007...	—	Проходная ставка: Не указано Предложите ставку
27.05	Кирпич асбоцементный 68 т	крытый Объем вагона: 138 м³	Аксенгер (рзд) (70... Май-Кудук (673806)	📅 30.05 📅 02.06	Проходная ставка: Не указано Предложите ставку
26.05	Плиты древесноволокн... 65 т	крытый Объем вагона: 158 м³	Приозерск (023002) Кызылсай рзд (6842...	—	Проходная ставка: Не указано Предложите ставку
26.05	Плиты древесноволокн... 65 т	крытый Объем вагона: 158 м³	Приозерск (023002) Канисай (рзд) (812...	—	Проходная ставка: Не указано Предложите ставку
26.05	Машины, оборудование... 60 т	крытый	Алма-Ата I (700007... Астана (690002)	📅 25.05	Проходная ставка: Не указано Предложите ставку

5.3 сурет. Тауарлар тізімі мен сүзгі

Кирпич асбоцементный

Груз Кирпич асбоцементный

Вес / Объем вагона: 68 т 138 м³

Дата перевозки: 📅 с 30.05.2016 по 02.06.2016

Тип вагона: крытый

Откуда: 📍 Аксенгер (рзд) (703804)

Куда: 📍 Май-Кудук (673806)

Отгрузок в месяц: Не указано

Проходная ставка: Не указано

KZT С учетом ЖД тарифа [+ Предложить ставку](#)

Идут торги

Создано:
27.05.2016 - 00:00

Похожие грузы для перевозки

- 📍 Аксенгер (рзд) (703804) - Май-Кудук (673806)
- 📍 Аксенгер (рзд) (703804) - Май-Кудук (673806)
- 📍 Аксенгер (рзд) (703804) - Астана (690002)
- 📍 Аксенгер (рзд) (703804) - Усть-Каменогорск (713702)
- 📍 Аксенгер (рзд) (703804) - Караганда-Сортировочная (673007)

5.4 сурет. Тауар беті

Тауар бетінде толық ақпаратпен қатар тауар статусы мен ұқсас тауралар тізімі көрсетіледі. Ұқсас тауарлар тасымалдау көлігі, жолы және уақыты бойынша іріктеледі.



Свободные подвижные составы

Фильтр		Сортировать по: дате добавления / дате актуальности		Актуальные / Не актуальные	
Тип вагона <input type="text" value="Введите тип ваго"/>		Дислокация <input type="text" value="Введите станцию"/>		Направление <input type="text" value="Введите станцию"/>	
Актуально <input type="text" value="с"/>		<input type="text" value="по"/>		<input type="button" value="Сбросить фильтр"/>	
04.12	крытый 68 т 158 м³ 1 шт	Алма-Ата I (700007) Алма-Ата II (700100)	04.12 11.12	50 KZT	
25.02	крытый 68 т 138 м³ 1 шт	Алма-Ата I (700007) Красноярская ж. д., РОССИЯ	25.02 03.03	12000 RUB	крытый
23.02	цистерна для спиртов 33 т 33 м³ 33 шт	Ангаты (раз) (998101) Куйбышевская ж. д., РОССИЯ	23.02 01.03	33 KZT	
13.02	крытый 68 т 138 м³ 1 шт	Алма-Ата I (700007) Брест-Центральный (130007)	13.02 20.02	12000 RUB	
13.02	крытый 68 т 138 м³ 1 шт	Алма-Ата I (700007) Брест-Центральный (130007)	13.02 20.02	12000 RUB	
13.02	крытый 68 т 138 м³ 1 шт	Алма-Ата I (700007) Брест-Центральный (130007)	13.02 20.02	12000 RUB	
12.02	ИВ-термос (переобору... 53 т 138 м³ 1 шт	Алма-Ата I (700007) Кичера (903401)	12.02 19.02	12000 RUB	
03.02	цистерна для спиртов 1 т 2 м³ 1 шт	Брест-Восточный (130204) Эперешке (эксп.) (000956)	03.02 10.02	12 KZT	

5.5 сурет. Көліктер тізімі

Оператор ақпаратты енгізіп қана қоймай, өңдей де алады. Ақпаратты өңдеу бетінде қажетті өзгертулерді жасауға болады. Өңдеу бетінде оператор туралы ақпаратты өзгертуге болады.

✎ Редактировать

Тип вагона *		
крытый		
Тоннаж	Объем	Количество вагонов
68 т	138 м ³	1 шт
Дислокация *		Желаемое направление *
Алма-Ата I (700007)		КИРГИЗИЯ
		Кыргызская ж. д.
		Каинды (715708)
Ставка за пользование *		
12000	RUB	
Актуально		
с	13/02/2016	
по	20/02/2016	
		

5.6 сурет. Өңдеу беті

6 Тіршілік қауіпсіздігі

Компьютерлер адам өмір салаларының барлығында дерлік қолданылады. Сондықтан біз электромагниттік өрістерден, дыбыстан және вибрациядан, иондаушы сәулелерден және басқа да қауіпті әрі денсаулыққа аса зиянды факторлардан зардап шегеді.

Еңбек пен тіршілік қауіпсіздігі ғылымы жұмысшыларға қауіпсіз жұмыскер орнын ұйымдастыру үшін туындады.

Еңбек қауіпсіздігі туралы ғылым жұмысшылардың денсаулығын қорғау, зиян, қауіпті әрі қолайсыз факторлардан қорғау әдістерінен тұратын, төтенше жағдайлар салдарын жоятын шаралар.

Тіршілік қауіпсіздігінің негізгі мақсаты мен мазмұны:

- Адам денсаулығына кері әсер ететін факторларды табу және зерттеу;
- Осы факторлардың салдарын қауіпсіз деңгейге дейін төмендету немесе барынша жою;
- Апаттың алдын алу және нәтижесін жою.

Адамзат өмірінің басты міндеттерінің бірі жұмыскерлердің денсаулығын сақтау, жұмыскер орнының қауіпсіздігін барынша қамтамасыз ету, кәсіби ауру-сырқау мен жаралану көзін жою болып келеді.

Жұмыскер орнында барлық мүмкін зиян және қауіпті өнеркәсіптік факторларды алдын алу шаралары жіті қарастырылуы тиімді. Факторлар деңгейі техникалық, заңдық және санитарлы-техникалық нормалар деңгейімен шектелген шамалардан аспау қажет.

Жобаның бұл бөлігі келесі сұрақтарды қарастырады:

- ДК операторының қолайлы еңбек жағдайларын анықтау;
- Жарықтандыру шығынын есептеу;

6.1 ДК операторының еңбек жағдайларын сипаттау және жұмыс бөлмесіне талаптар

Web-қосымшамен жұмыс жасау операторларға түсетін ақыл-ой және нейро-эмоционалдық жүктемелермен сипатталады. Мұнда пернетақтамен жұмыс жасаған кезде қол бұлшықеттері мен көзге түскен көп мөлшердегі қарбаласытық орын алады. Адам-оператордың қолайлы жұмыс қалпын ұстап тұруға тиімді жұмыскер орны бөлшектерінің орналасуы мен рационалды құрылымы үлкен әсер етеді.

Web-қосымшамен жұмыс істеу үрдісінде оңтайлы жұмыс және демалыс тәртібін ұстану тиімді. Қарсы жағдайда жұмыскерлерде бас, мойын, көз, арқа, бел және қол аурулары, ашуланшақтық, ұйқы бұзылуы, жұмыспен қанағаттанбаушылық пайда болауы мүмкін.

6.1.1 Кескіндеме және көрініс коэффициенттері

Жұмыс орны мен жиһаздың кескіндемелері көрнекі қабылдауға, көңіл-күйге оңтайлы жағдай туғызуы тиімді. Жарық сәулесі көздері болып табылатын шамдар мен терезелер экранға жарық сәулесі түсіре отырып, жұмыс жасауды қиындатады. Сол себепті, тоқтаусыз жұмыс жасайтын жұмыскерлерді стресске шалдықтырады. Жарық сәулесі көзінің шағылысуын минимумға жуықтандыру тиімді. Терезеден шағылысатын жарық сәулесі сәулесінен қорғану үшін әртүрлі перделер қолданылуы мүмкін.

Жарық сәулесі көздерінің орналасуына байланысты қабырға мен еденді келесі түстерге бояу тиімді:

- жарық сәулесі көздері оңтүстікке бағытталса:
- қабырғалар көгілдір немесе ашық көгілдір түске;
- еден жасыл түске;
- жарық сәулесі көздері солтүстікке бағытталса:
- қабырғалар сарғылт немесе сары сарғылт түске;
- еден қызыл-сарғылт түске;
- жарық сәулесі көздері шығысқа бағытталған:
- қабырғалар сары-жасыл түске;
- еден жасыл немесе сарғылт түске;
- жарық сәулесі көздері батысқа бағытталса:
- қабырғалар сары-жасыл немесе жасыл түске;
- еден жасыл немесе сарғылт түске.

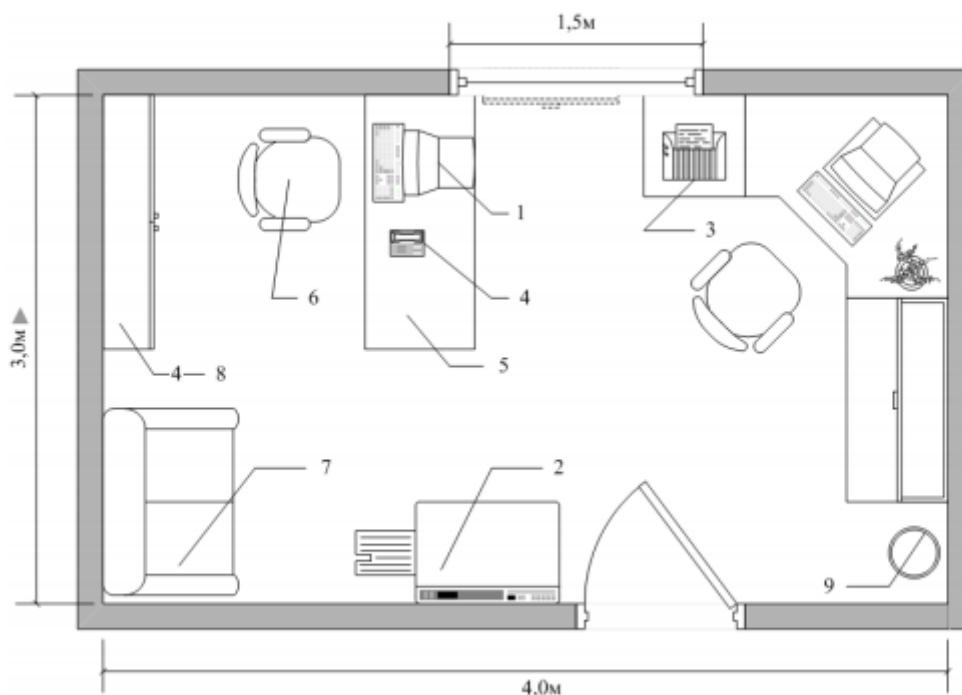
Компьютер орналасқан бөлмелерде шағылысу коэффициенті келесідей шамаларда болуы тиімді: төбе үшін: 60-70%, қабырғалар: 40-50%, едендер: 30% шамасында. Басқа беттер мен жұмыс жиһазы: 30-40%.

6.2 Ықтималды қауіпті және зиянды факторларды бағалау

Бөлмені талдау:

- жұмысшылар бөлмесі бірінші қабатта орналасады;
- жұмыс орны түрі: офис;
- бөлменің ұзынды: 4м, ені: 3м, биіктігі: 3м;
- күннен қорғаныс құралдары: жиналатын жалюзи;
- жарық сәулесі көзі 2 шам болады, әрқайсысы 2 люминесцентті лампа;
- бөлмеде бір терезе, бөлменің сол жағында орналасады, жарық көзі өлшемдері 2x1,5м;
- ішкі қабырғалар бояуы ашық түсті;
- бөлмеде күндізгі кезекте жұмыс істейтін 2 адам отырады.

Жұмыс күні 9:00-де басталып, 18:00-де аяқталады. Жұмыс орны сызбасы 6.1- суретте көрсетілгендей.



Сурет 6.1 - Офис. 1 - ДК, 2 - принтер/сканер, 3 - плот, 4 - телефондар, 5 - жұмыс үстелі, 6 - орындықтар, 7 - диван, 8 - шкаф, 9 - мусор тастайтын қоржын.

6.3 Дербес компьютермен жұмыс кезіндегі қауіпсіздік талаптары

ДК пайдаланушының денсаулығын сақтау және оның табысты жұмыс жасауы үшін оңтайлы жұмыскер орнын қамтамасыз ету талап етіледі. ДК пайдаланушының жұмыс қызметіне қолайсыз әсер ететін факторлар - видеодисплейлердің сәулеленуі. Компьютер терминалының сәулелені спектрі рентген, инфрақызыл ультракүлгін инфрақызыл және басқа да жиіліктегі электромагниттік толқындарының кең ауқымын қамтиды. Мамандар рентген сәулесінің зардабын тым аз деп санайды, себебі ол экранның заттарымен сіңіріледі. Ультракүлгін сәулелер қолданушы көзіне барлығынан артық әсер етеді. Ультракүлгін сәуленің көзге зиян әсері көздегі бөгде зат ретінде, көздің жасқа толуымен сипатталады. Дисплейдің артқы жағындағы сәуленің деңгейі зор. Сол себепті пайдаланушылар терминалдың артқы немесе шеткі беттерінен 1,2м қашықтықта, ал өз дисплейінен кемінде 0,5м қашықтықта орналасуы тиімді. Аз шаршау үшін жарықтығы кемінде 100кд/м^2 теріс контрастты дисплейді қолданған жөн. Өте жарық сәулесі жарықпен көзді шағылыстыру көзді тез шаршатады. Ол кәсіби ауруларға душар етуі мүмкін. Монохромды дисплей үшін жарқырауық нүкте 0,4мм-ден аспауы тиімді, түсті дисплей үшін 0,6мм-ден көп емес. Позитивтік контрастпен мәтінді өңдеу режимі кезінде бейненің қалпына келу жиілігі 72Гц-тен төмен болмауы тиімді. Бейненің $0,05 \div 1\text{Гц}$ аралықтағы төмен жиілікті тербелуі 0,1мм шамасында болуы қажет.

Пернетақта құрылысына арнайы талаптар қойылады. Ол дисплейге қатаң байланыспау тиімді. Пернетақтаның орналасу бұрышы $10 \div 15^\circ$ шамасында болуы тиімді. Компьютер құралдары бір түсті, күңгірт, шағылысу коэффициенті 0,4-0,6 болуы тиімді.

Компьютер қолданушылар жұмыскер орны туралы ұмытпайық.

Жұмыскер орны табиғи жарық сәулесі сол жақтан түсетіндей орналастырылуы тиімді. Жұмыс үстелі барлық тиімді қолданылатын жұмыс құралдарын сиғызу тиімді және оның биіктігі 680-800мм болуы тиімді.

Жұмыс орындығы (кресло) айналмалы-көтермелі болуы тиімді. Орындық (кресло) беті, арқасы және басқа бөліктері жартылай жұмсақ, сырғанақ емес, тоқ тудырмайтын және ластан сақтайтын, ауа өткізгіш болуы тиімді.

ДК операторы бөлмесінде алғашқы көмек қобдишасы мен өртсөндіргіш болуы тиімді.

Қалыпты жұмыс істеу үшін тағы да бір қажетті шарт жұмыскер орнында қалыпты метеорологиялық жағдай болуы. Микроклимат келесі параметрлермен анықталады: ауа температурасы, салыстырмалы ылғалдылық, жұмыскер орнындағы ауа жылдамдығы. Тиімдіті нормаларға сәйкес келмейтін тереологиялық шарт ДК пайдаланушының жұмысына және денсаулығына жаман әсер етеді.

Жарықтандырудың сапасы жоғары сипаттамасына жарықты көз алдына бірқалыпты етіп түсіру. Экранға және пернетақтаға шамшырақтан түсетін жарықты болдырмау үшін жарықтандыруға ерекше талаптар ұсынылады. Жарық сәулесі дақтары ақпаратты көзбен қабылдауға әсер етеді. Бұндау бөгеулер көзге салмақ түсіреді, зейін қоюды төмендетеді, тез шаршатады және ақпаратпен жұмыс кезінде қателіктер туғызады. Сол себепті көз алдындағы жарық сәулесі көздері $200\text{кд}/\text{м}^2$ -ден аспауы тиімді. Сол сияқты экрандағы жарық сәулесі $40\text{кд}/\text{м}^2$ -ден аспауы тиімді, шам жаққан кезде төбеден шағылысқан жарық сәулесі $200\text{кд}/\text{м}^2$ -ден аспауы тиімді.

Шамшырақтан түскен жарық сәулесі экранда жарық сәулесі дақтарын қалдырмауы тиімді. Жарық сәулесі шағылыспау үшін жұмыскер орны жарық сәулесі көздерден шетке қарай орналасуы тиімді. Егер жұмыскер орны жарық көзі қасында болса, дисплейді терезеге перпендикуляр орнату тиімді.

ДК оперторлары кей жағдайда табиғи жарықтың аздығынан немесе жетіспеушілігінен қатер шегеді. Жарықтың жетіспеушілігі көзді ғана емес, бүкіл ағзаны шаршатады.

Нақты жұмыс жағдайларында жарық сәулесі барлық жерге бірдей түспейді. Өртүрлі материалдардың жарық сәулесі коэффициенттері өртүрлі, сол себепті жарық сәулесі дақтары пайда болып, көрі қолайсыздығын тудырады. Көру қолайсыздығы зейінді нашарлатып, көз және жалпы шаршағандыққа алып соғады.

Компьютермен жұмыс кезінде тек табиғи дарық жеткіліксіз. Сол себепті біріктірілген жарықтандыру орындалады. Және де жасанды жарықты қараңғыда ғана емес, жарық сәулесі кезде де қолданады. Бір жақты жарықтандыру мүмкін емес, себебі терең және айқын көлеңкелер сияқты жағымсыз факторлар пайда болады.

Көбінесе жасанды жарық сәулесі үшін жарық сәулесі шашуы жоғары ($75\text{лм}/\text{Вт}$ және одан жоғары), қызмет мерзімі ұзақ (10 000 сағатқа дейін), табиғи жарыққа жақын люминесцентті шамадарды пайдаланады. Ең жарамдысы қуаты 80 Вт тең ақ түсті люминесцентті шамдар болып келеді.

6.4 Жасанды жарықты есептеу

Орташа дәлдіктегі және IV дәрежелі көзбен жұмыс кезіндегі жасанды жарықтағы жарықтандыру нормасы 300ЛК болып келеді. Жасанды жарықты тоқ көзімен істейтін жарық сәулесі арқылы алады, мысалы: люминесцентті шамдар және қызу шамдары.

Офисте люминесцентті шамадар қолданылған, себебі бұндай шамдардың артықшылықтары көп:

- спектральды құрамы бйынша бұндай шамдар табиғи жарық сәулесі көзіне жақын;
- ПӘК жоғары (қызу шамдармен салыстырғанда 1,5-2 есе жоғары);
- жарық сәулесі шашуы көтеріңкі сипатталады (қызу шамдарымен салыстырғанда 3-4 есе жоғары);
- жұмыс мерзімі едәуір ұзақ.

Төменде жұмыс орны үшін санау жүргізілген, өлшемдері: ұзындығы - 4 метр, ені - 3 метр, биіктігі - 3 метр.

Алдымен кабинеттің жарықтандырылуын санап, тұрақталған мәнімен салыстырайық. Бұл кабинеттегі жасанды жарықтандырудың жеткілікті екенін түсіну үшін тиімді (жарықтандыру жоспары 5.2-суретте келтірілген).

Шамшырақтың жұмыскер орны бетінен арақашықтығын есептейік

$$H = h - h_p - h_c \quad (6.1)$$

мұндағы h_c - шамшырақ пен жабынның арақашықтығы, $h_c = 0,05$ м;

h_p - жұмыскер орнының еденнен биіктігі, $h_p = 0,7$ м;

h - жұмыс орны биіктігі, $h = 3$ м.

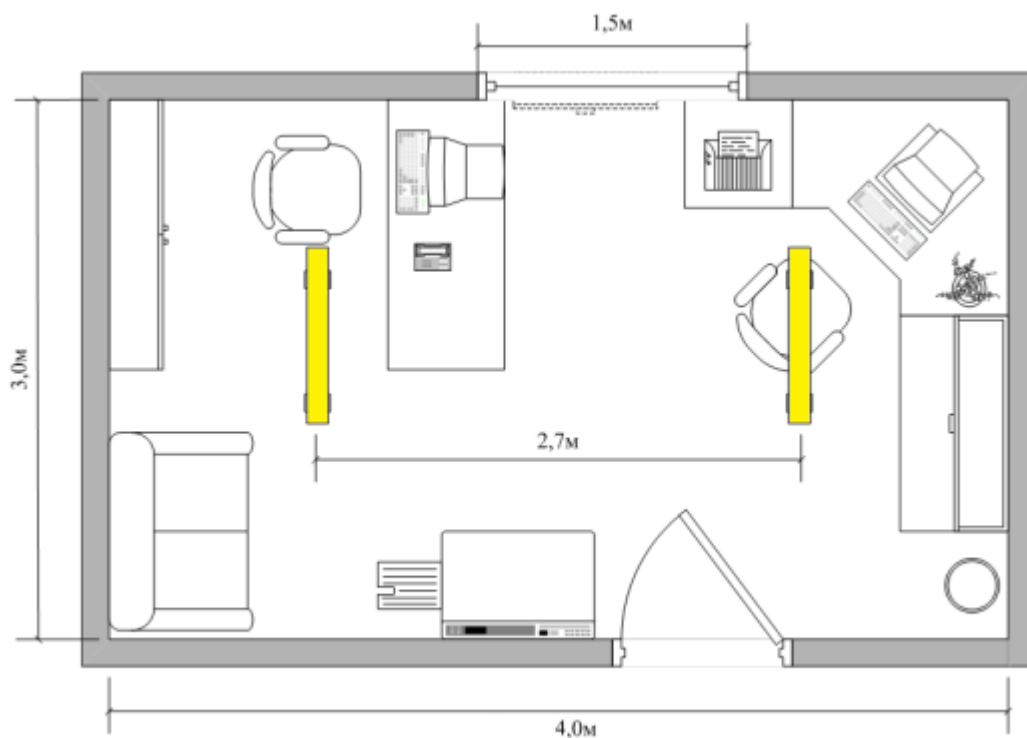
$$H = 3 - 0,7 - 0,05 = 2,25 \text{ м.}$$

Шамшырақтардың неғұрлым пайдалы арақашықтығын табайық, ол үшін формуласын қолданайық:

$$L = \lambda \times H \quad (6.2)$$

мұндағы λ - шамшырақтар арасында салыстырмалы энергетикалық және жарық-техникалық ең пайдалы арақашықтық ($\lambda = 1.2 \div 1.4$).

$$L = 1.4 \times 2.25 = 3,15 \text{ м.}$$



Сурет 6.2 - Офисты жарықтандыру жоспары

Жұмыс орны индексі (6.3) формуласымен есептеледі:

$$i = \frac{S}{H \times (A+B)} \quad (6.3)$$

мұндағы S - жұмыс орны ауданы, $S = 12 \text{ м}^2$;

H - аспаның есептеу биіктігі, $H = 2,25\text{м}$;

A - жұмыс орны ені, $A = 3\text{м}$;

B - жұмыс орны ұзындығы, $B = 5\text{м}$.

(6.3) формуласын қолдана отырып, аламыз:

$$i = \frac{12}{2,25 \times (3+5)} = 0,67$$

Жарықтандыруды есптеген кезде және бөлмені жоспарлаған кезде шағылысу коэффициенті сияқты коэффициент қолданылады. Бұл коэффициенттен бөлмені қажетті деңгейге дейін жарықтандыру кезінде қолданылатын жарықтандыру құралдарының сипаттамасы тәуелді.

Егер қабырғалардың, төбенің, еденнің шағылысу коэффициенті жоғары болса, онда аз қуатты шамшырақтар пайдаланылады.

Біздің жағдайда еден, қабырғалар, төбе шағылысу коэффициенттері тең болады (Кесте 5.1):

$$P_T = 50\%;$$

$$P_K = 30\%;$$

$$P_e = 10\%(\text{күңгірт түсті бет}).$$

Кесте 6.1 - Шағылысу коэффициенттері

Беттер сипаттамасы	Шағылысу коэффициенті, %
Ақ бет (ақ тұсқағаз/тақта, әк)	70
Жарық сәулесі бет (жарық сәулесі тұсқағаз, жарық сәулесі төбе, жарық сәулесі бояу)	50
Орта жарықты бет (жарық сәулесі паркет/линолеум, жарқын емес бояу/тұсқағаз)	30
Күңгірт түсті бет (күңгірт паркет/бояу/линолеум/тұсқағаз)	10

Жұмыс орны индексіне негізделе отырып, 6.2 кестесі арқылы қолдану коэффициентін анықтаймыз.

Кесте 6.2 - Люминесцентті шамды шамшырақ үшін қолдану коэффициенттер мәні, %

i	$P_T = 70\%;$ $P_K = 30\%;$ $P_e = 10\%$	$P_T = 50\%;$ $P_K = 30\%;$ $P_e = 10\%$	$P_T = 30\%;$ $P_K = 10\%;$ $P_e = 10\%$
0,5	28	21	18
1,0	49	40	36
3,0	73	61	58
5,0	80	67	65

Жұмыс орны индексі 0,67 болғандықтан, $\eta = 40\%$;

K_K - қор коэффициенті, $K_K = 1,2$.

Жарықтандыруды санау үшін (5.4) формуласын қолданайық:

$$E = \frac{N * \Phi_{ш} * \eta * n}{S * K_K * z}, \quad (6.4)$$

мұндағы E - көлденең жазықтықтың қажетті жарықтандыруы, Лк;

N - шамшырақтар саны;

$\Phi_{ш}$ - бір шамның жарық сәулесі ағыны, Лк;

η - жарықтандыратын қондырғының қолдану коэффициенті;

n - бір шамшырақтағы шам саны;

S - жұмыс орны ауданы, м²;

K_K - қор коэффициенті;

z - жарықтандырудың әркелкілік коэффициенті.

Жарықтандыру үшін біз ЛТБ 80 - 4 типті люминесцентті лампасын таңдап алдық, оның жарық сәулесі ағыны $\Phi_{ш} = 4300$ Лк.

Белгілі мәндерді (5.4) формуласына қойып, жарықтандыруды санайық:

$$E = \frac{2 \cdot 4300 \cdot 0,40 \cdot 2}{12 \cdot 1,2 \cdot 1,1} = 434,3 \text{ Лк}$$

Талапқа сай жарықтандыру 300 Лк тең, ал бөлмедегі жарықтандыру 434,3 Лк. Қорытынды: бөлмедегі жасанды жарықтандыру жеткілікті.

6.5 Табиғи жарықтандыруды талдамалы есептеу

Табиғи жарықтандыру деңгейін кенеттен аз уақыт аралықта және ауқымды көлемде өзгертуге болады. Сол себепті жұмыс орны ішінде табиғи жарықты нормалау негізгі шамасы табиғи жарықтандыру коэффициенті (ТЖК) болып саналады.

IV дәрежелі көру жұмыстары үшін ТЖК келесі формуламен есептеледі (6.5):

$$E_H = e_H \cdot mC, \quad (6.5)$$

мұндағы e_H - ТЖК мәні;

m - жарық сәулесі климатының коэффициенті, Кесте 6.3 бойынша жарық сәулесі ойысын бағдарлау үшін анықталады;

C - климаттың күн шуақтық коэффициенті (тура күн жарығын санағанда) 0,65 пен 1 аралығында өзгереді. Есептеу үшін климаттың күн шуақтық коэффициентін 0,9 деп алайық.

6.3 және 6.4 кестелеріне негізделе отырып ТЖК және жарықтық климат коэффициенті тең: $e_H = 1,5\%$; $m = 0,65$.

Кесте 6.3 - Табиға жарықтағы жарықтандыру нормалары

Орындалып жатқан жұмыс түрі	Ажырату объектінің өлшемі	Көру жұмысының дәрежесі	Табиғи жарық	
			ТЖК, e_H , %	
			Жоғарғы немесе біріккен жарық	Бүйірлік Жарықтандыру кезінде
Орташа дәлдікті	0,5 пен 1,0 аралығында	IV	4	1,5

Кесте 6.4 - Коэффициент мәндері

Жарық ойыстары сәулесі	Деңгей шеттеріндегі жарық ойыстаның бағыттары сәулесі	Жарық климатының сәулесі коэффициенті, m
		Администрациялық аудан тобының номері
		5
		Алматы

Құрылстың сыртқы кабырғаларында	С	0,75
	СШ, СБ	0,7
	Б, Ш	0,65
	ОШ, ОБ	0,65
	О	0,65

Менеджерлер кабинеті үшін ТЖК нормасын есептейік:

$$E_H = 1,5 * 0,65 * 0,9 = 0,88\%$$

Ары қарай жұмыс орны үшін ТЖК есептеп, нормамен салыстырайық. Кабинет өлшемдері: 4×3×3 сәйкесінше ұзындығы, ені, биіктігі. Жарық көзі еденне 0,8 м биіктікте орналасқан. Жарық көзі биіктігі - 2 м. (5.6) формуласы бойынша жарық көзі ауданын есептейміз:

$$S_T = \frac{S_n * e_H * \eta_T * K_F * K_K}{100 * \tau_{ж} * r_1} \quad (6.6)$$

(6.6) формуласынан e_H ашып көрсетейік:

$$e_H = \frac{S_T * 100 * \tau_{ж} * r_1}{\eta_0 * K_F * K_K * S_A} \quad (6.7)$$

мұндағы S_A - жұмыс орны ауданы, м²;

$$S_A = 4 * 3 = 12 \text{ м}^2.$$

S_T - кабинеттегі жарық көзі ауданы (жарық көзі өлшемдері 2×1,5), м²;

$$S_T = 2 * 1,5 = 3 \text{ м}^2.$$

мұндағы K_K - қор коэффициенті = 1,2 (қоғамдық және тұрғын ғимараттар бөлмелері үшін);

η_T - терезелердің жарық сәулесі сипаттамасы;

$\tau_{ж}$ - жарық сәулесі өткізгіштіктің жалпы коэффициенті.

(6.8) формуласы бойынша жарық сәулесі өткізгіштіктің жалпы коэффициентін есептейік:

$$\tau_{ж} = \tau_1 * \tau_2 * \tau_3 * \tau_4 \quad (6.8)$$

мұндағы τ_1 - материалдың жарық сәулесі өткізу коэффициенті: стеклопакет $\tau_1 = 0,8$;

τ_2 - жақтаудағы жарық сәулесі ойықтарында жоғалатын жарықты ескеретін коэффициент: қосарланған жақтау $\tau_2 = 0,6$;

τ_3 - көтергіш конструкцияда жоғалатын жарықты ескеретін коэффициент: темірбетонды құрылыс $\tau_3 = 0,8$;

τ_4 - жарықтан қорғаныс құралдарында жоғалатын жарықты ескеретін коэффициент (жиналатын жалюзи $\tau_4 = 1$).

Ондай болса:

$$\tau_{ж} = 0,8 * 0,6 * 0,8 * 1 = 0,384$$

Жұмыс орны ұзындығының тереңдігіне қатынасы:

$$\frac{L_6}{B} = \frac{4}{3} = 1,33$$

Үстелдің биіктігі еден бетінен 70 см, санитарлық ереже бойынша 2.2.2/2.4.1340-03- ЭЕМ-де жұмыс үшін арналған бір орынды үстел биіктігі, ал терезенің еденнен биіктігі 80 см және жарық көзі биіктігі 2 м болғандықтан, жұмыскер орны деңгейінен терезенің жоғарғы жағына дейінгі биіктік саналады:

$$(0,8 - 0,7) + 2 = 2,1 \text{ м} = h_1$$

Бөлменің ені және жұмыскер орны деңгейінің терезенің жоғарғы жағына дейінгі арақашықты қатынасы:

$$\frac{B}{h_1} = \frac{3}{2,1} = 1,43$$

6.5 таблицасынан $\eta_0 = 10,5$ шамасын аламыз

r_1 - жұмыс орны төбелерінен жарық сәулесі шағылуының арқасында бүйірлік жарықтың күшейуіндегі ТЖК өсуін ескеретін коэффициент.

Кесте 6.5 - Бүйірлі жарықтандыру кезіндегі η_0 жарық сәулесі сипаттамасының мәні

Жұмыс орны ұзындығының L_6 оның еніне B қатынасы	Кәдімгі жұмыскер орны деңгейіне h_1 жұмыс орны тереңдігінің биіктігіне қатысты жарық сәулесі сипаттамаларының мәні							
	1	1,5	2	3	4	5	7,5	10
4-тен жоғары	6,5	7	7,5	8	9	10	11	12,5
3	7,5	8	8,5	9,6	10	11	12,5	14
2	8,5	9	9,5	10,5	11,5	13	15	17
1,5	9,5	10,5	13	15	17	19	21	23
1,0	11	15	16	18	21	23	26,5	29
0,5	18	23	31	37	45	54	66	-

$$P_{op} = \frac{P_T + P_K + P_e}{3} = \frac{50 + 30 + 10}{3} = 30\%$$

Демек, 6.6 кестесіне негізделе отырып, $r_1=1,2$.

Енді K_F (терезелерге қарсы тұрған ғимараттардан түсетін көлеңкені ескеретін коэффициент) табайық:

$$\frac{L_F}{H_F} = \frac{15}{5} = 3,$$

мұндағы $L_F = 15\text{м}$ - қарсы тұрған ғимаратқа дейінгі қашықтық;
 $H_F = 5\text{м}$ - қарсы тұрған ғимараттың биіктігі.

Кесте 6.6 - r_1 коэффициентінің мәндері.

В/һ ₁	L/В	Төбе, қабырғалар, еденнің шағылысуының орташа коэффициенті Р _{ор}								
		0,5			0,4			0,3		
		L _б жұмыс орны ұзындығының В тереідігіне қатынасы								
		0,5	1	2 және жоғары	0,5	1	2 және жоғары	0,5	1	2 және жоғары
1-1,5 аралығында	0,1	1,05	1,05	1,05	1,05	1,05	1	1,05	1	1
	0,5	1,4	1,3	1,2	1,2	1,15	1,1	1,2	1,1	1,1
	1	2,1	1,9	1,5	1,8	1,6	1,3	1,4	1,3	1,2

6.7 кестеге негізделсек, $K_F = 1,0$. Барлық шамаларды (5.7) формуласына салайық:

$$e_H = \frac{3 \cdot 100 \cdot 0,448 \cdot 1,2}{10,5 \cdot 1,0 \cdot 1,2 \cdot 15} = 0,85\%$$

Кесте 6.7 - Терезеге қарсы тұрған ғимараттан түсетін көлеңкені ескеретін K_F коэффициент мәні

Қарастырылып жатақ және қарсы тұрған ғимараттардың арасындағы ұзындығы мен қарастырылған ғимараттың терезесінен қарсы тұрған ғимараттың төбесіне дейінгі биіктік қатынасы (L_F/H_F)	K_F
0,5	1,7
1,0	1,4
1,5	1,2
2,0	1,1
3 және одан жоғары	1,0

Норма бойынша ТЖК тең $E_H = 0,88\%$, іс жүзінде ТЖК $e_H = 0,85\%$. Олардың арасындағы айырмашылық тек 0,03% тең болғандықтан, жұмыс орны жарықтандыру нормаларына сәйкес жұмыскер орны деп саналады.

7 Экономикалық бөлім

7.1 Жоба сипаттамасы және негіздемесі

Бизнес-процесстердің оңтайлыны – кәсіпорын жұмысының басты экономикалық көрсеткіші екендігі белгілі. Демек, кіріс тауар тасымалдаудың өзіндік құнына тікелей байланысты. Тасымалдаушы кәсіпорынның негізгі міндеттері: тауар тасымалдауды оңтайландыру және бағасын азайту. Осы мақсатта іске асырылған жобаға қойылған талап: кәсіпорынның ақпараттық процесстерін автоматтандыруға арналған web-бағдарлама құру.

Төменде жобаны іске асырудың келесі экономикалық қыры қарастырылады:

1. web-қосымшаны іске асыратын программистерге төленетін қаржы (яғни қосымшаны іске асыруға қатысы бар тұлғалар) (J_a);
2. еңбекке ақы төлеу қоры ($EAK = J_a + Ш_{қос}$);
3. әлеуметтік салық($C_э$);
4. амортизациялық аударым (A);
5. қосымшаны іске асыруда қолданылатын материалдар (қағаз, картридж, канцелярлық тауарлар және т.б) (M);
6. интернет ақысы ($Ш_и$);
7. жұмыс орнының жалақысы, коммуналдық қызметтермен бірге ($Ш_A$);
8. және басқа да шығындар ($Ш_ө$). Сонымен, толық шығын келесі формуламен есептеледі:

$$C = EAK + C_э + A + M + Ш_и + Ш_A + Ш_{қос} \quad (4.1)$$

Біріншіден, web-қосымшаны іске асыруға қажет ресурс көлемін есептейік. Web-қосымшаны іске асыруға қажет ресурс көлемі 4.1-ші кестеде көрсетілген.

Кесте 7.1 - Web-қосымшаны іске асыруға қажет ресурс көлемі

Кезеңнің аталуы	еңбек көлемі, адам/сағ
Web-қосымшаны зерттеу	30
Web-қосымшаны құруға тапсырмалар жасау	40
Дизайн жасау	35
Деректер қорын құру	75
Код жазу	70
Бизнес-логиканы құру	30
Web-қосымшаны тестілеу	25
Web-қосымшаны қолданысқа енгізу	10
Қосындысы	315

Web-қосымшаны іске асыру үшін программист аптасына 5 күннен 7 сағаттан жұмыс жасайды. Сонымен 1.1 - кестесінен Web-қосымшаны жасауға 2

ай кететіндігін анықтадық. Бағдарламалық және аппараттық құралдарға кететін шығын 7.2 - кестеде көрсетілген.

Кесте 7.2 - Қосымша шығын

Атауы	Құны, теңге
Web-сервер жалға алу	130000
Web-дизайн	75000
Барлығы	205000

Қосымша жұмсалатын қаражат - (РПА) 205000 теңге.

Бір жылда жұмсалатын амортизациялық аударымды мына формуламен есептейміз:

$$A = A_{\text{бб}} \cdot N_A, \quad (7.2)$$

мұндағы A – амортизация соммасы;

$A_{\text{бб}}$ – аппараттың бастапқы бағасы;

N_A – амортизация нормасы.

Кесте 7.3 - Амортизациялық аударым

Атауы	Алғашқы құны, теңге	Амортизация нормасы, %	Бір жылдық амортизация соммасы, теңге
Web-сервер жалға алу	130000	30%	39000
Барлығы			39000

Бір жылға қажет амортизация қаржысы:

$$A = A_{\text{бб}} * 0,3 = 130000 * 0,3 = 39000 \text{ теңге}$$

1 еңбек ақысы амортизация қаржысы:

$$A = 39000 / 12 = 3250 \text{ теңге.}$$

2 айлық амортизация қаржысы:

$$A = 3250 * 2 = 6500 \text{ теңге.}$$

7.4 кестесінде жалақы қаржысы көрсетілген

Кесте 7.4 – Жалақыны қаржысын есептеу

Кезеңнің аталуы	Еңбек көлемі, адам/сағ	Бір сағат құны, теңге	Жиынтығы, теңге
Web-қосымшаны зерттеу	30	650	19500
Web-қосымшаны құруға тапсырмалар жасау	40	650	26000
Дизайн жасау	35	650	22750
Деректер қорын құру	75	650	48750
Код жазу	70	650	45500

Бизнес-логиканы құру	30	350	10500
Web-қосымшаны тестілеу	25	500	12500
Web-қосымшаны қолданысқа енгізу	10	250	2500
Қосындысы	315	188000	

Зейнетақы Аударымы (ЗА) жалақы қорының (ЕАҚ) 10% құрайды. Демек:

$$ЗА = ЕАҚ * 0,1 \quad ЗА = 188000 * 0,1 = 18800 \text{ теңге.} \quad (7.3)$$

Әлеуметтік салық (СӘ) айлықтың 11% - ін құрайды.

$$ОӘ = (ЕАҚ - ЗА) * 0,11 \quad ОӘ = (188000 - 18800) * 0,11 = 18612 \text{ теңге.} \quad (7.4)$$

Қосымша шығындар 7.5 және 7.6 кестелерінде берілген.

Кесте 7.5 - шығынды есептеу

Аталуы	Өлшем бірлігі	Бірлікке келетін баға, теңге	Барлығы	Ұзақтылығы	Қосындысы, теңге
Энергоқолдану (ноутбук)	0,09 (кВатт*с ағ)	19,59	37,35 (кВатт*с ағ)	315 сағ	556
Орынды жалға алу	м2	4000	12	2 ай	96000
Барлығы	96556				

Интернетке ақысы (РИ) айына 4000 тг. Сайтты жасауға 3 ай қажет, сонда ғаламтор ақысы РИ = 8000 теңге.

Кесте 7.6 - Қосымша шығынды есептеу

Атауы	Саны, дана	Бағасы, теңге	Қосындысы
Altel 4G	1	2000	2000
USB-жинақтаушысы	1	1000	1000
Қағаз	1	1500	1500
Қаламсап	3	50	150
Барлығы	4650		

Шығынды белгілі бер дәлдікпен ғана есептеуге болады. Қосалқы шығындарды есепке алалық – өндірістік шығыстар (Р_{ПР}). Бұл шығындар толық шығынның 20% болады.

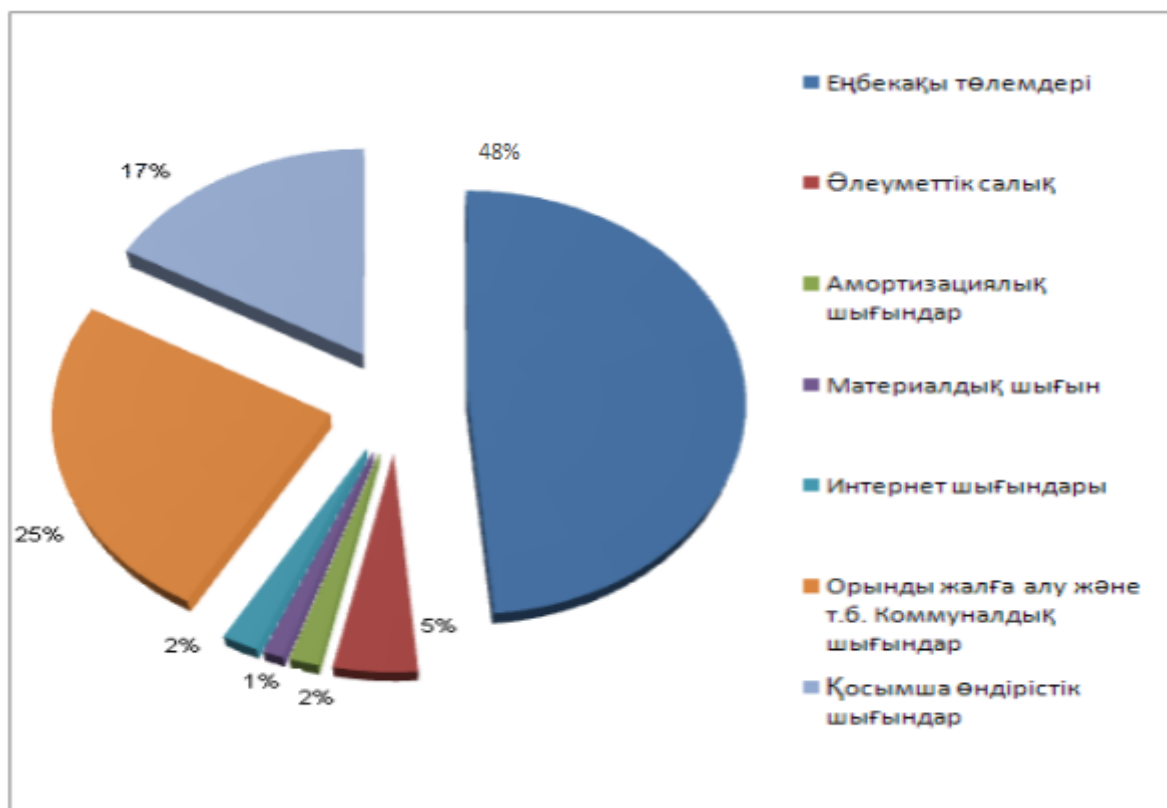
$$Р_{ПР} = (ЕАҚ + ОӘ + А + М + РИ + Ра) * 0,2 \quad (7.5)$$

$Р_{ПР} = (188000 + 18612 + 6500 + 4650 + 8000 + 96556) * 0,2 = 64464$ теңге мына Web-қосымшаның бағасы келесі формуламен есептеледі:

$$C = EAK + O_{\theta} + A + M + P_{И} + P_{A} + P_{Л} + P_{P} + P_{ПР} \quad (7.6) C$$

$$= 188000 + 18612 + 6500 + 4650 + 8000 + 96556 + 64464 = 386782 \text{ теңге}$$

7.1 диаграммасында web-қосымшаның бағасын құрайтын барлық шығындар келтірілген және әр бөліктің толық өзіндік құндағы үлесін анықтайық.



Сурет 7.1 – Web-қосымшаны іске асырудағы шығындар

Кесте 7.7 - Web-қосымша бағасы

Бөлшектік шығын	Соммасы, теңге	Толық өзіндік құнда алатын бөлігі, %
Айлық қоры	188000	48
Салық, 11%	18612	5
Амортизациялық аударым, %	6500	2
Қосымша шығын	4650	1
Интернетке ақысы	8000	2
Орынды жалға алу құны, коммуналдық қызметтермен бірге	96556	25
Қосымша өндірістік шығындар, 20%	64464	17
Барлығы	386782	

Экономикалық тиімділік web-қосымшаның құнын тапқанда, қажет рентабельдік деңгейді алайық, біздің жобада ол 40%.

$$ЦП = C(1+P/100), \quad (7.7)$$

ЦП = 386782 * (1 + 40/100) = 541495 теңге.

Жасалу бағасы қосымша құн салығымен (ҚҚС) бірге.

$$Ц_p = ЦП + ҚҚС \quad (7.8)$$

Қосымша құн салығы (ҚҚС) 2016 жылға қарағанда 12% құрайды.

Ц_p = 541495 * 1,12 = 606475 теңге.

7.2 Техничко-экономикалық тиімділігі

Автоматтандырылмаған ақпараттық жүйені қолданған кездегі шығын мен web-қосымшаны қолданған кездегі шығынды салыстыралық. Сондықтан жоғарыда есепке алынған шығын, АЖ-мен жұмыс жасайтын жұмысшылардың орташа айлық (120 000 тг), бір айдағы сағат мөлшері(176 с.), ноутбук бағасы керек. Осы көрсеткіштер жасалған web-қосымшаны қолдана іске асырылатын жұмысқа керек шығындарды есептеуге жеткілікті:

$$E_i = T_i \cdot (W/D + \frac{S \cdot (1+R)}{P_t}) \quad (7.9)$$

Бұл жерде, T_i – еңбекке жұмсалатын шығын;

W – web-қосымшамен жұмыс істейтін жұмысшының орташа жалақы мөлшері;

D – жұмыс күні;

S – web-қосымша бағасы;

R – ноутбук үшін төлемдер;

P_t – web-қосымша қызмет ету мерзімі.

Кесте 7.8 - Негізгі бизнес-процестерді орындау уақыты

Процес	Базалық нұсқа бойынша орындау уақыты, мин	Жобалық нұсқа бойынша орындау уақыты, мин
Негізгі ақпаратты енгізу	15	5
Тауар типтерін енгізу	10	5
Ақпаратты электронды түрге алмастыру	20	10
Сайтқа енгізу	7	3
Жаңа пайдаланушыларды тіркеу	15	10
Барлығы	67	33

Қазіргі қолданылатын жүйеде операцияларды орындауға кететін шығындар:

$$E_p = \frac{33}{60} * (\frac{100000}{176} + \frac{100000 \cdot (1+15)}{17500}) \approx 363 \text{ теңге}$$

Экономикалық тиімділікті E_p және E₀ айырмасы ретінде есептейміз. Бұл айырма 737 – 363 = 374 теңгеге тең. Бір пайдаланушы күніне осындай операциялардың 40 жуығы орындай алады. Сонымен, берілген жүйені енгізу ең кемінде $\frac{606475}{374 \cdot 40} = 40$ күннен кейін пайда әкеле бастайды.

Сәйкесінше жылдық (240 жұмыс күні) экономикалық тиімділік:

$$Y * 374 * 40 * 240 - 606475 = 2983925 \text{ теңге.}$$

Экономикалық тиімділік:

$$\frac{2983925}{606475} = 4.92$$

Есептеу нәтижелеріне сәйкес экономикалық артықшылық жоғары, бұл автоматтандырылған ақпараттық жүйе тауар тасымалдау процесін барынша оңтайландырады. Бұл құрылған өнімнің бағасын арттыруға бірден бір себеп. Бағдарламаны жұмыскерлер саны көп кәсіпорындарда қолдануға ұсынылады. Нарық бағалары бойынша ұқсас бағдарламалық құралдар орташа бағамен 500 000 теңгеге бағаланады. Бұл бағдарлама кәсіпорын жұмысын оңтайландыруға арналған.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР

- 1 А.К.Казешев. Сызықтық алгебра есептеуін бағдарламалау теориясы. - Алматы, 2006.
- 2 Ә.Б. Тунгатаров. Экономикалық мамандықтарға арналған арнайы курс. Алматы, 2005.
- 3 Е.Балапанов. Информатика терминдерінің қазақша орысша сөздігі. Алматы, 2000.
- 4 Аманжолов Ж. Охрана труда и техника безопасности.-Астана,2007.
- 5 Острейковский В.А. Информатика. – Обнинск: ОТАЭ, 2001.
- 6 Программирование на Visual C++ Глушаков С.В., Коваль А.В., Черепнин С.А. Худож.-оформ .А.С. Юхтман. - М.:ООО «Издательство АСТ»; Харьков: «Фолио», 2003.-726 с.
- 7 Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс/Худож.- оформитель А,С.Юхтман. – Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500 с.
- 8 Баженова И.Ю. Visual C++6.0 (VISUAL STUDIO 98). Уроки программирования. - М.: Диалог-МИФИ, 2001. -416 с.
- 9 Под ред. Хомоненко А.Д. Основы современных компьютерных технологий. – СПб: Корона-Принт, 2000.
- 10 Йодан Э. Структурное программирование и конструирование программ. - М.: - Мир, 2007.
- 11 Макарова Н.В. Информатика: Учебник – М.: Финансы и статистика, 2006.
- 12 Харви Дейтел, Пол Дейтел, Как программировать на C++: Третье издание. Пер. с англ.-М.: ЗАО «Издательство БИНОМ», 2001. -1152 с.: ил.
- 13 Гук М. Аппаратные средства РС. –СПб: Питер, 2004.
- 14 Могилев А.В. Новейшее технологий XXI века.-СПб: Питер, 2004.
- 15 Бобарыкин В. А., Прудовский Б. Д., Трофимова Г. И. Новые модели и методы решения задач оптимального использования транспортных средств. М.: «Транспорт», 2001.
- 16 Биллиг В. Основы программирования на C#: Учеб. пособие – СПб.: Питер, 2006., 346с.
- 17 <http://ru.wikipedia.org>.
- 18 Красс М. С. Математика для экономических специальностей. М.: ИНФРА-М, 2002.
- 19 Просветов Г. И. Математические методы и модели в экономике: Задачи и решения. М.: Издательство “Альфа-Пресс”, 2008.
- 20 Сапарбаев. Ә. Ж., Ахметов. К. А., Мақұлова. А. Т. Экономикалық-математикалық әдістер мен модельдер. Оқулық – Алматы: «Қазақстан жоғарғы оқу орындар қауымдастығы». 2005.
- 21 Сапарбаев. Ә. Ж., Ахметов. К. А., Мақұлова. А. Т. Математикалық программалау. Оқулық – Алматы: «LEM». 2001.

- 22 Кузнецов А. В., Сакович В. А., Холод Н. И. Высшая математика: Математическое программирование. Мн.: Вышэйшая школа, 2001 г.
- 23 Иванилов Ю.П. Математические модели в экономике. М-Наука,2004.
- 24 Калинина В.Н. Математическая статистика .М – 2000.
- 25 Справочник по математике для экономистов (Под редакцией Ермакова.- М. Высшая школа,1987.)
- 26 Статическое моделирование и прогнозирование.(Под редакцией А.Г. Гранберга) –М. Финансы и статистика,2005.
- 27 А. Құсаинов «Қазақша – орысша, орысша – қазақша терминологиялық сөздік». Алматы, Рауан, 2003.
- 28 Акулич.И.Л. математическое программирование в примерах и задачах. – М. Высшая школа, 2006.
- 29 Воробьев Н.Н. Теория игр для экономистов – кибернетиков. – М. Наука, 2000.

ҚОСЫМША А

Wagons.php

```
<?php namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Support\Facades\DB;
```

```
class Wagon extends Model {
```

```
    protected $fillable = ['id_crm', 'wagon_type', 'weight', 'size', 'count', 'author',  
    'location', 'direction', 'price', 'currency', 'tariff', 'actual', 'from_date', 'to_date'];
```

```
    public function getWagonType(){  
        $wagon = DB::table('wagon_types')->find($this->wagon_type);  
        if($wagon)  
            return $wagon->text;
```

```
        return 'Не определён';
```

```
    }
```

```
    public function getStation($station_id){
```

```
        $station_arr = explode(';', $station_id);
```

```
        if(count($station_arr) > 1) {
```

```
            $text = ";
```

```
            if($station_arr[0]) {
```

```
                $station = DB::table('train_stations')->find($station_arr[0]);
```

```
                if($station && $station->code != 'YTM001') // != Любое
```

```
                    $text .= $station->text . ' , ';
```

```
            }
```

```
            if($station_arr[1]) {
```

```
                $road = DB::table('roads')->find($station_arr[1]);
```

```
                if($road)
```

```
                    $text .= $road->name . ' , ';
```

```
            }
```

```

        if($station_arr[2]) {
            $country = DB::table('countries')->find($station_arr[2]);

            if($country)
                $text .= $country->name . ', ';
        }

        return $text ? substr($text, 0, -2) : 'Не определён';
    }

    $station = DB::table('train_stations')->find($station_id);

    if($station)
        return $station->text;
    return 'Не определён';
}

public function getSimilarGood($all = null){
    $wagon = $this;
    $total = [];

    $similar = Good::where('from_station', '=', $wagon->location)-
>orWhere(function($query) use ($wagon){
        $query->where('to_station', '=', $wagon->direction);
    }->get());

    foreach($similar as $s) {
        $total[] = $s;
    }
    if(!$total)
        return null;

    usort($total, function($a, $b) use ($wagon){
        $a_score = $b_score = 0;
        if($a->from_station == $wagon->location) $a_score++;
        if($b->from_station == $wagon->location) $b_score++;

        if($a->to_station == $wagon->direction) $a_score++;
        if($b->to_station == $wagon->direction) $b_score++;

        if($a_score == $b_score) return 0;

        return ($a_score < $b_score) ? 1 : -1;
    });
}

```



```

    });

    if($all) {
        return $total;
    }

    return array_slice($total, 0, 5);;
}
}

```

Goods.php

```
<?php namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
```

```
class Good extends Model {
```

```
    protected $fillable = ['id_crm', 'date_create', 'etsng', 'wagon_type', 'weight',
'wagon_size', 'from_station', 'to_station', 'price', 'currency', 'tariff', 'from_date',
'to_date', 'shipping_count'];
```

```
    public $timestamps = false;
```

```
    public function getEtsng(){
        $etsng = DB::table('etsng')->find($this->etsng);
        if($etsng)
            return $etsng->text;
    }

```

```
        return 'Не определён';
    }

```

```
    public function getWagonType(){
        $wagon = DB::table('wagon_types')->find($this->wagon_type);
        if($wagon)
            return $wagon->text;
    }

```

```
        return 'Не определён';
    }

```

```
    public function getStation($station_id){
        $station = DB::table('train_stations')->find($station_id);
    }

```

```

        if($station)
            return $station->text;
        return 'Не определён';
    }

    public function getSimilarGood($all = null){
        $good = $this;
        $total = [];

        $similar = Good::where('from_station', '=', $good->from_station)-
>where('id', '!=', $good->id)->orWhere(function($query) use ($good){
            $query->where('to_station', '=', $good->to_station)
                ->where('id', '!=', $good->id);
        }->get());

        foreach($similar as $s) {
            $total[] = $s;
        }
        if(!$total)
            return null;

        // sorting similar goods from from_station and to_station
        usort($total, function($a, $b) use ($good){
            $a_score = $b_score = 0;
            if($a->from_station == $good->from_station) $a_score++;
            if($b->from_station == $good->from_station) $b_score++;

            if($a->to_station == $good->to_station) $a_score++;
            if($b->to_station == $good->to_station) $b_score++;

            if($a_score == $b_score) return 0;

            return ($a_score < $b_score) ? 1 : -1;
        });

        if($all) return $total;

        return array_slice($total, 0, 5);
    }
}

```

GoodController.php

```
<?php namespace App\Http\Controllers;
```

```

use App\Http\Requests;
use App\Http\Controllers\Controller;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Auth;
use App\Good;
use DateTime;
use App\Events\RateWasAdded;
use Event;

class GoodController extends Controller {

    public function __construct()
    {
        $this->middleware('auth');
    }

    public function addRate(Request $req){
        $inputs = $req->all();
        $good = Good::find($inputs['good_id']);
        if(!$good)
            return redirect()->back()->withErrors('Труз не найден...');

        if(isset($inputs['tariff']))
            $inputs['tariff'] = 1;
        else
            $inputs['tariff'] = 0;

        $inputs['user_id'] = Auth::user()->id;
        unset($inputs['_token']);

        $soap = Auth::user()->getSoap();

        $oldRate = DB::table('rates')->where('user_id', '=', $inputs['user_id'])->where('good_id', '=', $inputs['good_id'])->get();
        if($oldRate) {
            $inputs['updated_at'] = new DateTime();

            DB::table('rates')->where('id', '=', $oldRate[0]->id)->update($inputs);
        }
    }
}

```

```

        if($soap) {
            $soap->set_price([
                'id_crm' => $good->id_crm,
                'currency' => $inputs['currency'],
                'price' => $inputs['rate'],
                'tariff' => $inputs['tariff'],
                'author' => Auth::user()->id_crm
            ]);
        }

        return redirect()->back()->with('message', 'Ставка изменена!');
    }

    $inputs['created_at'] = new DateTime();

    DB::table('rates')->insert($inputs);

    if($soap) {
        $soap->set_price([
            'id_crm' => $good->id_crm,
            'currency' => $inputs['currency'],
            'price' => $inputs['rate'],
            'tariff' => $inputs['tariff'],
            'author' => Auth::user()->id_crm
        ]);
    }

    // Notifying user
    Event::fire(new          RateWasAdded($good->getSimilarGood(true)
,$inputs));

    return redirect()->back()->with('message', 'Ставка добавлена!');
}

public function getRate($user_id = NULL, $good_id = NULL, $is_not =
NULL){

    $rate = DB::table('rates');

    if($user_id){
        if($is_not == NULL){
            $rate = $rate->where('user_id', '=', $user_id);
        } else {
            $rate = $rate->where('user_id', '!=', $user_id);
        }
    }
}

```

```

        }
    }

    if($good_id)
        $rate = $rate->where('good_id', '=', $good_id);

    $rate = $rate->get();

    return $rate;
}

/**
 * Display a listing of the resource.
 *
 * @return Response
 */
public function index(Request $request)
{
    if ( !$request->ajax() )
    {
        return view('pages.good.goods');
    }

    // for localhost show all goods
    (env('APP_ENV') == 'local') ? $actual_date = 0 : $actual_date = time() -
(31 * 24 * 60 * 60);

    $goods = Good::where('date_create', '>=', date('Y-m-d', $actual_date))-
>get();
    $rates = $this->getRate(Auth::user()->id);

    return view('pages.good.goods-ajax', ['goods' => $goods, 'rates' =>
$rates]);
}

/**
 * Show the form for creating a new resource.
 *
 * @return Response
 */
public function create()
{
    //
}

```

```

/**
 * Store a newly created resource in storage.
 *
 * @return Response
 */
public function store()
{
    //
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($id)
{
    $good = Good::find($id);
    if(!$good)
        return redirect()->route('goods');

    $similar = $good->getSimilarGood();

    $rate = $this->getRate(Auth::user()->id, $id);

    if($rate)
        $rate = $rate[0];

    $other_rates = $this->getRate(Auth::user()->id, $id, TRUE);

    return view('pages.good.getGood', ['good' => $good, 'rate' => $rate,
'other_rates' => $other_rates, 'similar' => $similar]);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function edit($id)
{

```

```

        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param int $id
     * @return Response
     */
    public function update($id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return Response
     */
    public function destroy($id)
    {
        //
    }
}

```

WagonController.php

```

<?php namespace App\Http\Controllers;

use App\Http\Requests;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;

use App\Http\Requests\StoreWagonRequest;
use App\Wagon;
use DateTime;
use Illuminate\Http\Request;
use Auth;
use App\Events\WagonWasAdded;
use Event;

class WagonController extends Controller {

```

```

public function __construct()
{
    $this->middleware('auth');
}

/**
 * Display a listing of the resource.
 *
 * @return Response
 */
public function index(Request $req)
{
    $wagons = Wagon::where('author', '=', Auth::user()->id);
    $status = "";
    if($req->input('s') != 'old'){
        $wagons = $wagons->where('actual', '!=', 0);
        $status = 'new';
    } else {
        $wagons = $wagons->where('actual', 0);
        $status = 'old';
    }

    $wagons = $wagons->get();

    return view('pages.wagon.wagons', ['wagons' => $wagons, 'status' =>
$status]);
}

/**
 * Show the form for creating a new resource.
 *
 * @return Response
 */
public function create()
{
    $countries = DB::table('countries')->get();
    $roads = DB::table('roads')->get();
    return view('pages.wagon.addWagon', ['countries' => $countries, 'roads'
=> $roads]);
}

/**
 * Store a newly created resource in storage.
 *
 * @return Response

```



```

*/
public function store(StoreWagonRequest $req)
{

    $inputs = $req->all();

    if(isset($inputs['tariff']))
        $inputs['tariff'] = 1;
    else
        $inputs['tariff'] = 0;

    $from_date = new DateTime('NOW');
    $to_date = new DateTime(date("Y-m-d H:i:s", mktime(0, 0, 0,
    $from_date->format('m'), $from_date->format('d') + intval($inputs['actual']),
    $from_date->format('Y'))));

    $inputs['from_date'] = $from_date;
    $inputs['to_date'] = $to_date;
    $inputs['author'] = Auth::user()->id;

    $inputs['id_crm'] = '0';

    $anyStation = DB::table('train_stations')->where('code', 'YTM001')-
>first();

    if( !isset($inputs['direction']) ){
        $inputs['direction'] = $anyStation->id;
    }

    $directionRoad = isset($inputs['directionRoad']) ? DB::table('roads')-
>where('id', $inputs['directionRoad']->get() : null;
    $directionCountry = isset($inputs['directionCountry']) ?
DB::table('countries')->where('id', $inputs['directionCountry']->get() : null;

    $wagon = Wagon::create($inputs);

    $wagon->direction .= ';';

    if($directionRoad) {
        $wagon->direction .= $directionRoad[0]->id;
    }

    $wagon->direction .= ';';

    if($directionCountry) {

```

```

        $wagon->direction .= $directionCountry[0]->id;
    }

    $wagon->save();

    Event::fire( new WagonWasAdded( $wagon->getSimilarGood(true),
    $wagon ) );

    $soap = Auth::user()->getSoap();

    if($soap){
        $wagonType      =      DB::table('wagon_types')->find($wagon-
    >wagon_type);
        $wagon->wagon_type = $wagonType->code;

        $location = DB::table('train_stations')->find($wagon->location);
        $wagon->location = $location->code;

        if($directionCountry) {
            $wagon->direction = $directionCountry[0]->name . ';';
        } else {
            $wagon->direction = ';';
        }

        if($directionRoad) {
            $wagon->direction .= $directionRoad[0]->name;
        }
        $wagon->direction .= ';';

        $direction = DB::table('train_stations')->find($inputs['direction']);
        $wagon->direction .= $direction->code;

        $wagon->from_date = $wagon->from_date->format('m.d.Y');
        $wagon->to_date = $wagon->to_date->format('m.d.Y');

        $wagon->author = Auth::user()->id_crm;

        $id_crm = $soap->set_train($wagon);
        $wagon->id_crm = $id_crm->return;

        DB::table('wagons')->where('id', '=', $wagon->id)-
    >update(['id_crm' => $id_crm->return]);
    }

```

```

        return redirect()->route('getWagon', $wagon->id);
    }

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($id)
{
    $wagon = Wagon::find($id);

    if(!$wagon || $wagon->author !== Auth::user()->id)
        return redirect()->route('wagons')->withErrors(['У вас нет
привелегии']);

    $similar = $wagon->getSimilarGood();

    return view('pages.wagon.getWagon', ['wagon' => $wagon, 'similar' =>
$similar]);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function edit($id)
{
    $wagon = Wagon::find($id);
    if(!$wagon || $wagon->author !== Auth::user()->id)
        return redirect()->route('wagons')->withErrors(['У вас нет
привелегии']);

    $countries = DB::table('countries')->get();
    $roads = DB::table('roads')->get();
    return view('pages.wagon.editWagon', ['wagon' => $wagon, 'countries'
=> $countries, 'roads' => $roads]);
}

/**
 * Update the specified resource in storage.
 *

```

```

* @param int $id
* @return Response
*/
public function update(StoreWagonRequest $req)
{
    $wagon = Wagon::find($req->input('wagon_id'));
    if(!$wagon || $wagon->author !== Auth::user()->id)
        return redirect()->back()->withErrors(['У вас нет привелегии']);

    $inputs = $req->all();

    if(isset($inputs['tariff']))
        $inputs['tariff'] = 1;
    else
        $inputs['tariff'] = 0;

    unset($inputs['wagon_id']);
    unset($inputs['_token']);

    $to_date = date_create_from_format('d/m/Y', $inputs['actual']);
    $inputs['to_date'] = $to_date;

    unset($inputs['actual']);

    $anyStation = DB::table('train_stations')->where('code', 'YTM001')-
>first();

    if( !isset($inputs['direction']) ){
        $inputs['direction'] = $anyStation->id;
    }

    $directionRoad = isset($inputs['directionRoad']) ? DB::table('roads')-
>where('id', $inputs['directionRoad'])->get() : null;
    $directionCountry = isset($inputs['directionCountry']) ?
DB::table('countries')->where('id', $inputs['directionCountry'])->get() : null;

    $inputs['direction'] .= ';';

    if($directionRoad) {
        $inputs['direction'] .= $directionRoad[0]->id;
    }

    $inputs['direction'] .= ';';

```

```

        if($directionCountry) {
            $inputs['direction'] .= $directionCountry[0]->id;
        }

        $wagon->update($inputs);
        $wagon->save();
        return redirect()->route('getWagon', $wagon->id)->with('message',
'Изменения сохранены');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return Response
     */
    public function destroy($id)
    {
        $wagon = Wagon::find($id);
        if(!$wagon || $wagon->author !== Auth::user()->id)
            return redirect()->back()->withErrors(['У вас нет привелегии']);
        $soap = Auth::user()->getSoap();
        if($soap){
            $resp = $soap->delete_train(['id_crm' => $wagon->id_crm]);
        }

        $wagon->delete();

        return redirect()->route('wagons')->with('message', 'Вагон удален');
    }
}
}

```

ProfileController.php

```

<?php namespace App\Http\Controllers;

use App\Http\Requests;
use App\Http\Controllers\Controller;

use Illuminate\Http\Request;
use App\Http\Requests\UpdateProfileRequest;

```

```
use App\Http\Requests\UpdatePasswordRequest;
use App\Http\Requests\UpdateEmailRequest;
use Auth;
use Hash;
```

```
class ProfileController extends Controller {
```

```
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Display a listing of the resource.
     *
     * @return Response
     */
    public function index()
    {
        return view('pages.account.profile', ['user' => Auth::user()]);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @return Response
     */
    public function store()
    {
        //
    }

    /**
```

```

* Display the specified resource.
*
* @param int $id
* @return Response
*/
public function show($id)
{
    //
}

/**
* Show the form for editing the specified resource.
*
* @param int $id
* @return Response
*/
public function edit()
{
    return view('pages.account.editProfile', ['user' => Auth::user()]);
}

/**
* Update the specified resource in storage.
*
* @param int $id
* @return Response
*/
public function update(UpdateProfileRequest $req)
{
    $inputs = $req->all();

    $tPhone = "";
    foreach($inputs['phone'] as $p) {
        $tPhone.= $p . '|';
    }
    $inputs['phone'] = substr($tPhone, 0, -1);

    if(isset($inputs['subscribe']))
        $inputs['subscribe'] = 1;
    else
        $inputs['subscribe'] = 0;

    $user = Auth::user();
    $user->update($inputs);
    $user->save();
}

```

```

        return redirect()->route('editProfile')->with('message', 'Изменения
сохранены');
    }

    public function updateEmail(UpdateEmailRequest $req) {
        $user = Auth::user();

        if( ! Hash::check($req->input('password'), $user->password))
            return redirect()->route('editProfile')->withErrors(['Неверный
пароль']);

        $user->email = $req->input('email');
        $user->save();

        return redirect()->route('editProfile')->with('message', 'Email
изменена');
    }

    public function updatePassword(UpdatePasswordRequest $req) {
        $user = Auth::user();

        if( ! Hash::check($req->input('oldPassword'), $user->password))
            return redirect()->route('editProfile')->withErrors(['Неверный
пароль']);

        $user->password = bcrypt($req->password);
        $user->save();

        return redirect()->route('editProfile')->with('message', 'Пароль
изменена');
    }
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return Response
 */
public function destroy($id)
{
    //
}
}

```


ApiController.php

```
<?php namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

use App\Http\Requests;
use App\Http\Controllers\Controller;
use App\User;
use App\Good;
use App\Wagon;
use Mail;
use App\Services\Mailers\AppMailer;

class ApiController extends Controller {

    public function getTrainStations(Request $req){
        $q = $req->input('q');

        if(!$q){
            return json_encode([[ 'id'=>', 'text'=>'Начните вводить
станцию']] );
        }

        $data = DB::table('train_stations');

        return json_encode($data->select('id', 'text')->where('text', 'LIKE', '%'.
$q.'%')->take(10)->get());
    }

    public function getWagonTypes(Request $req){
        $q = $req->input('q');

        if(!$q){
            return json_encode([[ 'id'=>', 'text'=>'Начните вводить тип
вагона']] );
        }

        $data = DB::table('wagon_types');
        return json_encode($data->select('id', 'text')->where('text', 'LIKE', '%'.
$q.'%')->take(10)->get());
    }

    public function addUser(Request $req){
```

```

    $user = User::create([
        'id_crm' => $req->input('id_crm'),
        'company' => $req->input('company'),
        'email' => $req->input('email'),
        'password' => bcrypt($req->input('password')),
        'subscribe' => 1
    ]);
    return 'ok';
}

public function addGood(Request $req){
    $inputs = $req->all();

    $etsng = DB::table('etsng')->where('code', '=', $inputs['etsng'])->first();
    if(!$etsng)
        $etsng = DB::table('etsng')->where('code', '=', "000000")->first();
    $inputs['etsng'] = $etsng->id;

    $wagon_type = DB::table('wagon_types')->where('code', '=',
    $inputs['wagon_type'])->first();
    $inputs['wagon_type'] = $wagon_type->id;

    $from_station = DB::table('train_stations')->where('code', '=',
    $inputs['from_station'])->first();
    $inputs['from_station'] = $from_station->id;

    $to_station = DB::table('train_stations')->where('code', '=',
    $inputs['to_station'])->first();
    $inputs['to_station'] = $to_station->id;

    $soldGood = Good::where('id_crm', '=', $req->input('id_crm'))->first();

    if($soldGood){
        $good = $soldGood->update($inputs);
        $soldGood->save();
    } else {
        $good = Good::create($inputs);
    }

    return 'ok';
}

public function deleteGood($id){
    $good = Good::where('id_crm', $id)->first();

```

```

        if(!$good) return 'not found good with id ' . $id;

        DB::statement('SET FOREIGN_KEY_CHECKS = 0');

        DB::table('rates')->where('good_id', $good->id)->delete();

        $good->delete();

        DB::statement('SET FOREIGN_KEY_CHECKS = 1');

        return 'ok';
    }

    public function checkWagonsActuality(AppMailer $mailer){
        $wagons = Wagon::where('actual', 1)->where('to_date', '<', date('Y-m-
d'))->get();

        foreach($wagons as $wagon) {
            $wagon->actual = 0;
            $wagon->save();

            $user = User::find($wagon->author);

            $mailer->wagonActuality($wagon, $user);
        }

        return "Actuality changed: " . count($wagons);
    }

    public function getStationsFromRoads($road_id) {

        $stations = DB::table('train_stations')->where('road_id', '=', $road_id)-
>get();

        return json_encode($stations);

    }

}

```