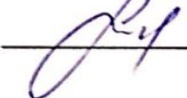




Университет	Батыс Қазақстан университеті
Названия	Метрис:ЕвгенийТессолядиплом.росс
Автор	Метрис:Евгений
Координатор:	Алия Адильбекова
Дата отчета:	2019-05-30 10:34:02
Коэффициент подобия № 1: <input checked="" type="checkbox"/>	28,3%
Коэффициент подобия № 2: <input checked="" type="checkbox"/>	17,5%
Длина фразы для коэффициента подобия № 2: <input checked="" type="checkbox"/>	25
Количество слов	4 929
Число знаков:	40 347
Адреса пропущенные при проверке:	
Количество завершенных проверок: <input checked="" type="checkbox"/>	76

Ғылыми жетекші  А. К. Адильбекова

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Мамандығы 5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету

Студент Жетпіс Бекболат

Тақырыбы: «Android платформасындағы Онлайн Автобөлшек мобильді қосымшасын құру»

### ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Дипломдық жобада күнделікті кездесетін және адам өмірі-не тереңінен еніп жатқан компьютерлік техника, ақпараттық жүйе мен желінің дамуы ақпарат көлемінің жылдам өңделуіне жақсы әсер етуде. Қазіргі таңда мобильді қосымша және көлікке арнайы қосымшалар көптеген түрлері қолдануда.

Дипломдық жоба көлік қолданушылары үшін және алыстан бөлшек келмеуі үшін қолайлы және қағаз жұмысын жеңілдетуге мүмкіндік беретін мобильді қосымша дайындаған.

Дипломдық жұмыс кіріспеден, үш бөлімнен, қолданылған әдебиеттер тізімінен, қорытындыдан тұрады. Бірінші бөлімде мәселенің қазіргі замандағы жағдайына шолу және оны талдау жасалынды. Екінші бөлімде программалау тілдерін пайдалану ерекшеліктері жобаланған. Үшінші бөлімде қосымшаның қолданушылар үшін бөлшектердің таңдау корпоративті бөліктері мен жасауды дайындалған.

Дипломдық жобаны қарастыра келгенде жақсы жақтарын айта кетуге болады. Жұмыс материалы жоспарлы түрде, қойылған талаптарға сай, тақырып төңірегінде жазылған. студенттің жұмысты рәсімдеудегі біліктілігін, талдаушылық қабілетін көруге болады. Осы жұмыста негізгі өзекті мәселелерді шеше отырып, зерттеу жұмысында оңтайлы нәтижелер алып, жұмысты қойылған талаптарға сай орындап шықты.

Қортындылай келгенде, осы зерттеу жұмысын орындау барысында Жетпіс Бекболат Қанатұлының айтарлықтай жақсы еңбектенгені көрі-неді.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Жетпіс Бекболат Қанатұлының, 5В070400– «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ

«Программалық инженерия»

кафедрасының лекторы «30»



А.К. Адильбекова

04

2019 жыл

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Жетпіс Бекболат Қанатұлы

«Android платформасындағы Онлайн Автобөлшек мобильді қосымшасын құру»  
мобильді-қосымшасы

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

5B070400 – «Есептеу техникасы және бағдарламалық камтамасыз ету»  
мамандығы

Алматы 2019



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,

ассистент-профессор

 Р. Юнусов

" 30 " мае 2019 ж.

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

Тақырыбы: «Android платформасындағы Онлайн Автобөлшек мобильді қосымшасын құру» мобильді-қосымшасы


5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Орындаған

Б. Қ. Жетпіс

Ғылыми жетекші

техн. ғыл. магистрі, лектор

 А.Адилбекова

" 30 " 04 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

**БЕКІТЕМІН**

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,  
ассистент-профессор

 Р. Юнусов

" 27 " маусым 2019 ж.

**Дипломдық жоба орындауға  
ТАПСЫРМА**

Білім алушы: Жетпіс Бекболат Қанатұлы

Тақырыбы: «Android платформасындағы Онлайн Автобөлшек мобильді қосымшасын құру» мобильді-қосымшасы

Университет академиялық мәселелер жөніндегі проректорының бұйрығымен «14» наурыз 2018ж. № 1841-б шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі «13» мамыр 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Ұсынылатын дипломдық жобаны құру барысында силлабусты автоматты түрде құрастырып беретін конструкторлық мобильді қосымша жасау

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) аналитикалық шолу;

б) жобалау бөлімі;

в) өңдеу құралдары және жобаны талдау;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма.

е) Б Қосымшасы – бағдарлама мәтіні.



Жобаның презентациялық 12 слайды ұсынылған.

Ұсынылған негізгі әдебиеттер: 13 атау.

**Дипломдық жобаны орындау  
КЕСТЕСІ**

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Диплом жұмысының жоспар-құрылымын құру.	14.01.2019	жоқ
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	жоқ
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім беру бойынша есеп беру жазбасын дайындау	01.02.2019	жоқ
4. Дипломның екінші бөлімі – жобалау сызбаларын дайындау.	15.02.2019	жоқ
5. Жобаның мобильді қосымшасын тестілеуден өткізу.	18.03.2019	жоқ
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2019	жоқ

**Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары**

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Таурбекова А. А. лектор	04.05.19	
Бағдарламалық бөлім	Қалдыбеков С. Б. сениор-лектор	06.05.19	

Ғылыми жетекші \_\_\_\_\_ Адильбекова А.

Тапсырманы орындауға қабылдап алған студент \_\_\_\_\_ Жетпіс Б.Қ

Күні \_\_\_\_\_ «03» маусым 2018 ж.



## АҢДАТПА

Қазіргі кезде барлық адамдар мобильді телефонды өте көп қолданады. Бүгін, адамдардың өміріне ұялы телефон рөлін асыра бағалау қиын. Кез келген уақытта осы ықшам құрылғымен бізді қызықтыратын ақпаратты табуға, әріптестерін, олардың туыстары мен достары байланыстыра алуға болады. Барлығы ноутбук, ұялы телефонды әрдайым қолданады.

Клиенттермен жұмыс істеу тәжірибесі өте маңызды және көбінесе бизнесті құру және интернеттегі бөліктерді сату үшін маңызды. Электронды коммерция шешімін таңдағанда, бақытты тұтынушыны есте ұстану керек. Үлгі мобильді қосымшасын басып, «сатып алуды» жасау арқылы сатып алушының орнына қою керек. Есте сақтай отыра, тұтынушы бөлігін тауып, сатып алу үшін көбірек қадамдар жасаса, ол сатуға бас тартатын ықтималдығы көбірек болады.

Қосымша кез-келген адамға арналған, сондықтан қосымша жоғарғы технологияларға сәйкес болуы керек, тартымды дизайнға және ыңғайлы интерфейске ие болуы керек, және де, көпфункционалды, кез-келген қосымша-қолданушыға ыңғайлы болуы керек.

Жоба бойынша, Android мобильді қосымшасы клиент ретінде, ал қосымша-бет сервер ретінде, клиент-сервер технологиясын іске асырады.

REST сервис тек қана Android платформасына ғана емес, кез-келген құрылғыға, яғни Android, IOS, web-бет, Windows операциялық жүйелерінің барлығына веб-бетті қолдануға мүмкіндік береді.

## АННОТАЦИЯ

В настоящее время практически все люди пользуются мобильным телефоном. Сейчас, невозможно оценить важность мобильного телефона. В любое время суток в мобильном телефоне можно поговорить с родными, найти нужную информацию. Все пользуются ноутбуками и мобильными телефонами.

Опыт работы с клиентами очень важен и особенно важен для создания бизнеса и продажи частей в интернете. При выборе решения электронной коммерции необходимо помнить счастливого потребителя. Шаблон необходимо вставить на место покупателя, нажав на мобильное приложение и создавая «покупку». Помните, что потребитель будет делать больше шагов для того, чтобы найти и купить часть, он будет иметь больше вероятности отказаться от продажи.

Приложение направлена на большую аудиторию, поэтому приложение должна быть высоко производительным, многофункциональным, должен иметь привлекательный дизайн, должен быть удобен любому пользователю.

По проекту клиентская часть разработана на Android платформе, а серверная часть разработана как веб-сайт.

Rest сервис может не только обслужить подобных операционных систем, он так же может обслужить как IOS систем, так же и Windows операционных систем.



## ANNOTATION

Currently practical all people use a mobile phone. Now, it's impossible to appreciate the importance of a mobile phone. At any time of the day in your mobile phone you can talk with your family, find the information you need. All use laptops and mobile phones.

Experience with clients is very important and particularly important for the creation of business and sale of parts on the Internet. When choosing an e-Commerce solution, it is necessary to remember the happy consumer. The template must be inserted into the buyer's place by clicking on the mobile app and creating a "purchase". Remember that the consumer will take more steps in order to find and buy the part, he will be more likely to refuse the sale.

The application is aimed at a large audience, so the site should be highly productive, multifunctional, must have an attractive design, should be convenient to any user.

Under the project, the client chat is developed on the Android platform, and the server part is designed as a application.

Rest service can not only serve similar operating systems, it can also serve both iOS systems, as well as Windows operating systems.

## КІРІСПЕ

Осы заманауи кезеңде мобильді қосымшаның мүмкіншілігі шексіз. Әрі көптеген проектілер өзекті түрі деректерді ыңғайлылықпен оңай жолда басқаруға, меңгеруге арналған даму заманы болып табылады. Проектінің қызметі мобильдік қосымшаға тұрақты және ыңғайлы мобильді қосымша құру болып көрсетіледі. Бгі таңда мобильді телефон мобильді қосымшалар индустриясының қарқынды дамуы арқасында жұмыс істейтін, бос уақытты және білім алу үшін әртүрлі және қажетті ресурстардың көзі болып табылады. Қазақстанда сондай-ақ тиісті бағдарламалық қамтамасыз етуді дамытуда белсенділік бар, сондықтан нарықты жан-жақты талдауға, негізгі ойыншыларды анықтауға және үрдістерді болжауға қажеттілігі туралы айтуға болады.

“Онлайн Автобөлшек” проектісінің түпкі мақсаты, кез-келген жеке көлігі бар адамдарға өз тауарын немесе қызметін ұсынуды барынша жеңілдету және жылдамдату. Проектінің жүзеге асырылуында бар салынған ақпараттар әділ түрде сатып алушыларға жетеді. “Онлайн Автобөлшек” проектісінде қолданылатын технологиялар:

- Android операционды мобильдік системасы;
- REST технологиясы;
- Java бағдарламалық тілі;
- MySQL.

Онлайн Автобөлшек проектісі керекті бөлшекті оңай іздеуге, белгілі бір бөлшекке байланысты шынайы сын-пікірді айқын жариялауға, тең бәсекелестікті жүргізуге арналған. Онлайн Автобөлшек проектісі мобильді қосымшадан тұрады. Мобильді қосымша мен бет бет арасындағы байланыс REST сервис, JSON форматы және PHP сұраныстары арқылы жүзеге асырылады. Осы уақытқа дейін адамның бөлшек-құрылғысына керек етуін қолжетімділігіне байланысты, ыңғайлылығын өңдеу уақыты қысқартылып, көлікке керекті бөлшекті іздеу барысында пайда болған қателер саны азаяды. Кәсіпорынның үнемі нәтижесінде компьютердің көмегімен ақпарат көлемі мен құжат айналымын өңдеу қиынға соғуда.

Бүгінгі таңда бұл міндеттер Интернет-ресурстармен жұмыс істеу, музыка тыңдау, фотосуреттер түсіру, ойындар мен қосымшаларды пайдалану арқылы толықтырылды және де магазин қорымен керекті зат болмаса нарсені адамның ыңғайлылығына қарай толықтай дамыған. Болашақтың ұтқырлығы, ең бастысы, теледидарды қарау, әртүрлі жабдықтарды бақылау, оның иесінің денсаулық жағдайын бақылау функцияларын және, әлдеқайда көп, негізгі міндеттер тізіміне қосылады. Қолданбалар қолданба интерфейсінің бағдарлама кодынан қолданбалы логиканың бағдарламалық кодын бөлу қағидасы негізінде жасалады.

Бұл мобильді қосымшаның негізгі мақсаты бөлшектік құрал-жабдықтардың сатылымы болып келеді, әрі әрбір телефон қолданушыға жеткізумен және керекті жабдықтарын дер уақытында алып келу қызметі

болып табылады. Қазақстанда жалпы Жапония елінің танымал көліктері көп қолданысқа ие автокөліктердің бірі, олардың танымалдылары Toyota мен Chevrolet. Осы модельді көліктер қолжетімді болғандықтан, бөлшектердің мобильді қосымша арқылы сатылымы мен жеткізу қызметін жүзеге асыру.

Ал негізгі міндеттерінде қосымша арқылы тек екі типті модельді көліктердің бөлшектердің сатылым жасап, әрі тез уақыт аралығында жеткізу болады. Әрбір қолданушының жіберген тапсырыстарын админ қабылдап жұмыс барысын қадағалап отырады. Берілген кодтардың тікелей байланысы REST-пен беріліп java арқылы қабылдап алынады.

Онлайндық автокөліктерді сатуға киім, кітап немесе зергерлік бұйымдарды сатумен салыстыру мүмкін емес. Интернеттегі автомобиль өнеркәсібі өте ерекше қажеттіліктерге ие, сондықтан автомобиль индустриясына арналған арнайы электрондық коммерция шешімі қажет.

Ортақ платформалар оны кеспейді. Олар сенімді OEM каталогын ұсына алмайды, және бұл бөліктер мобильді қосымшаға қатысты өте қажет.

Бөлшектерге арналған электрондық коммерция нарығы жыл сайын өсіп келеді және әрбір адам нарыққа үстемдік етуі керек. Адамдарға қандай жүйенің олар үшін жақсы екенін және дилердің жетістігін анықтауға көмектесу үшін, автобөлшектерді желіде сатуға арналған дұрыс электрондық коммерция шешімін қалай табуға қатысты тәжірибелік нұсқаулықты әзірлеу керек.

## 1 Негізгі бөлім

### 1.1 Жобаның өзектілігі

Қазіргі заманда мобильді қосымша дамуына байланысты, адамның екінші бір маңызды рөлін атқарады және күрделі мәселелерді шешетіндей ыңғайлылығымен адамның жағдайына зор көмектеседі. Мобильді қосымша, көп функционалды, лезде іздеуден бастап, әрі керекті модельді көліктің керекті бөлшегін табуға болады. Мобильді қосымша аудиториясы барлық адамдарға әсіресе ер адамдарға арналған. Өйткені, әр қалыпта көлік жағдайына байланысты тез арада бір бөлшек керек болады, сол кезде тіркелусіз кіре алып керекті бөлшегін таба алады. Дипломдық жобаның өзектілігі көпфункционалды іздеу типтерінің бар болуында, және мобильді қосымшамен REST технологиялық ауқымын кеңінен қолдана отырып байланысқа түсе алуында. Дипломдық жоба андроид құрылғысымен жазылған, қазіргі кезде кеңінен тарап келе жатқан тіректердің бірі болғандықтан, жоба көп уақыт бойы қолайлы басқаруға ие болады.

Қазіргі кезде барлық адамдар мобильді телефонды өте көп қолданады. Бүгін, адамдардың өміріне ұялы телефон рөлін асыра бағалау қиын. Кез келген уақытта осы ықшам құрылғымен бізді қызықтыратын ақпаратты табуға, әріптестерін, олардың туыстары мен достары байланыстыра алуға болады. Барлығы ноутбук, ұялы телефонды әрдайым қолданады. Телефонда сақталған контактілер, сонымен қатар көптеген тағы басқа да ақпарат көп - және несиелік карта нөмірлерін, күнделікті естелік, түрлі файлдар, адамдардың идеялары мен ойлары жазылады. Яғни телефон адамға дереу қойын кітапшасын және флэш-дискісін ауыстырады. Қазіргі заманда ұялы телефон шағын бола отырып, дербес компьютердің орнын баса алмаса да, мүмкіндігі түймедей кем түспейді. Ал, мобильді қосымшаны қолдану адамға, ұйқы немесе тамақтану сияқты әдеттегі іс қимылдардың біріне айналды. Соған байланысты, қазір күніне мыңдаған мобильді қосымшалар жарыққа шығып жатыр. Әрқайсысы, неше түрлі бағытқа бағытталған, бірі адамдардың тілдесуіне болса, ал, бірі жаңалық оқуға арналған, соған қарай көлікте тікелей қатысты болып келеді. Оның ішінде ыңғайлылыққа арналғандары да кездеседі, мысалы, керекті бөлшекті тез табу секілді. Сондықтан, «Автобөлшек» проектісінің түпкі мақсаты автокөлік иелеріне керекті құрылғыларын тез әрі қолжетімді іздеуді қолайлы ету. Жоба мобильді қосымша мен құралады. Мобильді қосымша деректер қорын веб-сайттан алады. Жергілікті немесе ұлттық деңгейдегі бөлшектерді қалай сатқыңыз келсе, он-лайн бөлшектер бизнесіңізді басқарып, басқаруға байланысты.

Ұлттық деңгейде төлемдер жоғары болуы мүмкін, бірақ ол да бәсекеге қабілетті. Бұл сондай-ақ жоғарыда қалу үшін көп уақыт пен күш-жігерді білдіреді.



Сіз сондай-ақ интернет дүкен арқылы бөліктерді сайтқа сату мүмкіндігіңіз бар. Егер сіз әлі де осы онлайн-бизнеске сенімді болмасаңыз, сіз өзіңіздің веб-сайтыңызға кіріп жатқан аймақтағы тұтынушыларға көңіл бөлесіз, содан кейін оларды меңгерген кезде ел бойынша кеңейте аласыз.

Ақырында, Автодок және Exist сияқты басқа сату арналары да кейбір артықшылықтарға ие. Қандай шешім қабылдасаңыз да, өз қызметтеріңізді қалағаныңызға бейімдейтін электрондық коммерция шешімін табылады. Олар қазіргі мақсаттарға жауап бере алады, сонымен бірге болашақ мақсаттарымызға қол жеткізу үшін ауқымды болуы керек.

## **1.2 Ұқсас мобильді қосымша жобаларын шолу**

Қазіргі уақытта мобильді қосымшасы автокөлік, автобөлшектер карамағында қарқынды дамып келе жатыр. Android жүйесіндегі Play Market мобильді қосымша дүкенінен әрбір әзірлеушінің жасалған қосымшасын тапса болады. Соның бірі Автодок – 1998 жылдан бастап Ресейдегі автокөліктердің жетекші сатушысы. Бүгінгі таңда бұл жетекші әлемдік өндірушілердің автомобиль өнімдерін сатуға көшбасшылардың бірі болып табылады. Сыни қараса бұл компанияның ауқымды әрі қызмет ету белгісі жақсы ірі компания болып келеді. Алдағы уақытта талдап әрі зерттеп бақылай отыра озат мамандармен бірлесіп ортақ жоспарды іске асырса болады.

Автодок компаниясы Kayaba, Total, Liqui Moly, Motul, Sangsin Brake және тағы басқа көптеген танымал брендтермен жұмыс істейді. Компанияның баға тізімдері 3000 сауда маркасының 22 млн. Күн сайын электронды каталогтарын жаңартады, сондықтан үнемі жаңартылған ақпаратты алуға және қажетті қосалқы бөлшектерді уақытты іздеусіз бір жерде таба алуға болады.

Бұл компанияның қосымшасына електей отыру оңайға соқпайды,әлі алдағы уақыттарда өзімнің мобильдік қосымшаның тікелей дамуына бастапқы әрекеттердің бірі болып санаса болады. Автодок негізгі ресейлік автоөндірушілердің (VAG, Toyota, Nissan, Mercedes Benz, Форд, Мобис және басқалар) орталық қоймалары, ірі автокөліктердің көтерме қоймалары, сондай-ақ, БАӘ, Жапония АҚШ, Германия және Корея тікелей байланысы бар.

Автодоктың әрбір клиенті тапсырыс берушіге тапсырысты орналастырудан бүкіл тізбекті бақылайтын жеке менеджерге ие болады. Егер сіз таңдау бойынша шығынға ұшырасаңыз, онда жеке менеджер сіздердің барлық сұрақтарыңызға жауап бере алады, өндірушілер туралы қажетті ақпаратты ұсынып, таңдауға көмектеседі. Салыстырмалы түрде біздің мобильді қосымша осыған ұқсастығы бар, колл-центр ретінде хабарласа отырады. Сондай-ақ, автомобильіңіздің VIN-кодының бөліктерін таңдау туралы сұранысты әрдайым қалдыра аласыз және біздің мамандар сізге қысқа уақыт ішінде көмекке келеді.

Осы компанияда қызметкерлердің жоғары кәсібилігі және әрбір клиентке жеке көзқарасы клиенттерге жоғары сапалы бөлшектермен қамтамасыз етуге және логистикалық міндеттемелерін қатаң сақтауға мүмкіндік береді.

### 1.3 REST сервистер құрылымы

Мерзімді RESTful функцияға қолдану ғана болса, қызмет REST шарттарын қолданып, жүзеге асыра алады. Сондықтан, осы принциптер, REST бізге мынадай мүмкіншілігін бере алу үшін, REST сәулеті қарапайымдылығы негізгі артықшылығы деп айтуға болар еді. Ой-пікірдің қарапайымдылығы, қарапайым дизайн және RESTful қосымшаларға функционалдық қосуға болады. Бұл, ең алдымен түсіну тым қиын болғандықтан, сондықтан, қарапайым және әмбебап құрал болып табылады. Біз, қазіргі заманның өзінде бағдарламашылар тобы көп қабатты немесе кез келген жаңа қабатын қосу емес, бірақ жай ғана пайдалану ұзақ стандарттарын таныды. Сондықтан, артықшылықтары мен кемшіліктері туралы сұраққа жауап, талдау және одан да көп мағынасын жасады деп, SOAP және REST тәсілдердің көбірек қолдануға ұсынылған.

SOAP осы өңдеу тұрғысынан ауыр және күрделі параметр болып табылады, бұл тікелей оның, хаттамалар мен стандарттар жиынының отбасы болып табылады.

REST негізделген, ал SOAP, көлік протоколы ретінде HTTP-ді пайдаланады. Бұл SOAP технологиясы басқа да құралдарын іздеу қажет, ал сервер деңгейінің пысықтауы осындай кэш ретінде HTTP протоколына негізделген барлық қолданыстағы шешімдер, сол REST сәулетінің жұмыс істеу жалғастырады. Оның орнына, SOAP қызметі көлік орнына HTTP туралы кез келген хаттамаға жұмыс істеу мүмкіндігі ретінде мифтік мүлікті қабылдайды, бірақ оның практикалық қолдануы өте сирек кездеседі. Бұл, осы технологияның іске асырылуының қиындылығымен анықталады.

RESTful қызметтерді дамыту әлдеқайда оңай деп саналады. Мүмкін, бұл бастапқы даму ортасы ретінде, бірақ біздің тамаша бағдарламалық қамтамамызды іске асырылатын мүмкіндігі бар «блокнот» бағдарламасын пайдаланса болады, бірақ бұл мүмкіндіктің өзектілігі күман туғызуы мүмкін болып табылады.

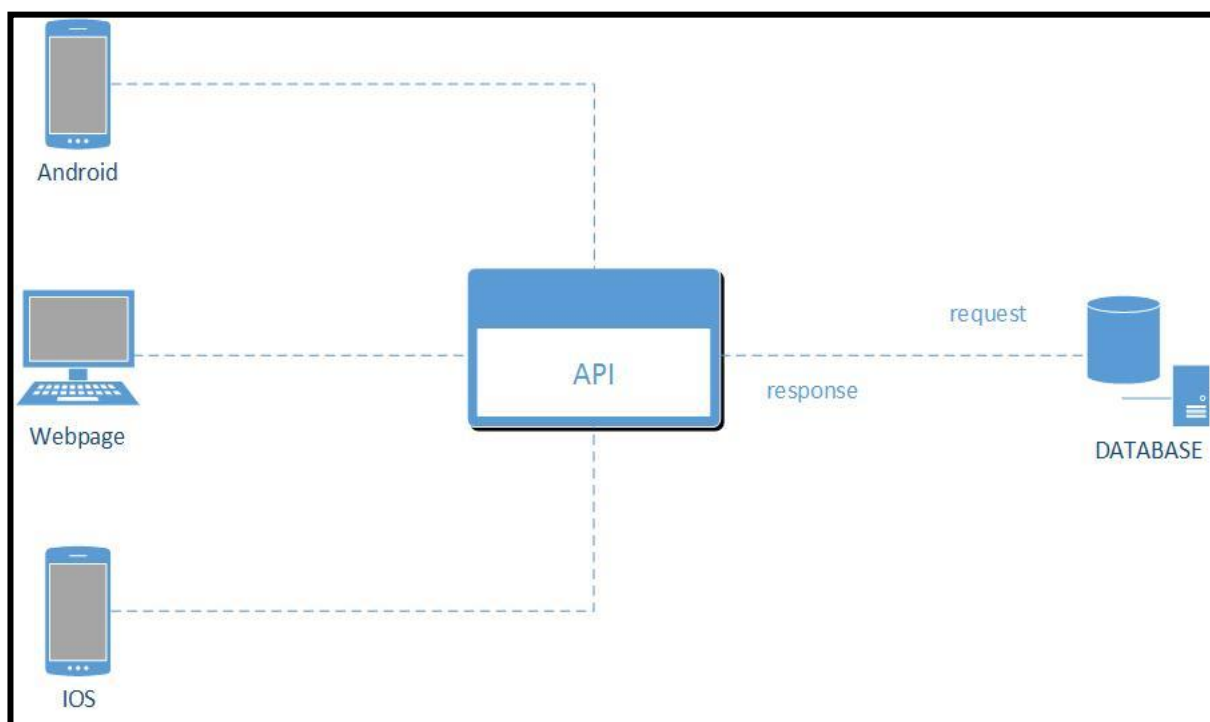
«SOAP» үшін алғашқы Google қорытындысы REST жауабы түрлі форматтарда ұсынылуы мүмкін, және SOAP XML міндетті екеніне баса назар аударады. Бұл шын мәнінде JavaScript бағдарламасымен бастап, қызметін енгізу маңызды фактор, өте міндетті болып қарастырылады, біз белгілі JSON алуға келетін жауап болып табылады.

«SOAP немесе REST» әлі күнге дейін барлық осындай тамаша жобалау құралдарын пайдаланып, веб қызметті толық сипаттау үшін WSDL хаттама бар. SOAP құрылымын өте қиындатылып жасалған.

REST (Representational State Transfer) – желіде таратылатын бағдарлама компоненттері арасында өзара іс-қимыл сәулет стилі. REST үлестірілген гипермедиялық жүйелерін жобалау кезінде ескеріледі. Шектеулері келісілген жиынтық болып табылады. Белгілі бір жағдайларда (онлайн дүкендер, іздеу жүйелері, және басқа да жүйелер деректер негізінде), бұл өнімділігі жоғары технология мен оңайлатылған сәулет әкеледі.

Интернетте, қашықтан сұраныс жіберу рәсімдері, кәдімгі HTTP-сұрау болуы мүмкін, және қажетті деректер сұрау параметрі ретінде беріледі.

Осы технологияның барлық шектеу мен шарттарын қарастыра отырып жасаған технология «RESTful» деп те аталады. REST технологиясы 1.1-суретінде көрсетілген.



**1.1-сурет – REST технологиясы**

SOAP принципіне негізделген веб-қызметтің айырмашылығы, RESTful Android-API үшін ешқандай «ресми» стандарт жоқ. REST технологиясы URL, JSON және XML стандарттарды қолданады. Барлық сұраныстарға, яғни, мәлімет алу, мәлімет енгізу, мәліметті жою және мәліметті жаңарту функциялары қарастырылған REST жүйесі API деп аталады.

Restful веб сервистерін құру, бағдарламалаудағы басқа білімдер сияқты, өнердің бір ұшы, ғылымның бір ұшы болып табылады. Веб-жобалау бағдарының дамуына байланысты, REST сервисті құрудың мүмкіншіліктері көбеюде, және тәжірибелік жұмыстарда көбеюде. REST сервисті құрудың анықталған заңдылықтары жоқ, HTTP протоколын санамағанда, REST сервисті ыңғайлы етіп жасаған және HTTP протоколын толықтай қолданған клиент жағынан API-ді қолданатын клиенттер үшін дұрыс болады.

REST архитектурасы алты шектеумен анықталады. Бұл шектеулер ең алғаш Рой Филдингтің докторлық диссертациясында жазылған. Алты шектеу:

- бірегей интерфейс;
- күйдің болмауы;
- жауапты кәштеу;
- клиент-сервер;
- көпдеңгейлі жүйе;
- мұқтаждық бойынша жазылған код.

Бірыңғай интерфейс (Uniform Interface). Бірегей интерфейс клиент пен сервердің арасындағы интерфейссті анықтайды.

Жалпы, әр бөліктің жеке дамуына көмектесетін архитектураны жеңілдетеді және бөлектейді. Бірыңғай интерфейсстің төрт принципі:

Белгілі бір ресурстар сұраныста анықталады, URI болса, ресурстардың идентификаторы ретінде жүреді. Клиентке қайтатын көріністерге қарағанда ресурстар онцептуалды түрде жекеленген. Мысалы: сервер өзінің деректер қорын жібермейді, керісінше, XML, JSON түрінде кейбір деректерді, немесе, клиент сұратқан мәліметтерді осы форматтарда жібереді.

JSON үшін қысқа және ақпаратты ұйымдастырылған және қарапайым түрде сақтау тәсілі. Қысқаша айтқанда, бұл бізге нақты логикалық жолмен қол жеткізе алатын деректердің оқуға болатын жиынтығын береді.

Көріністер арқылы ресурстарды басқару. Қолданушы ресурс туралы, және метамәлімет туралы кішкене ақпаратқа ие болса, ол деректер қорындағы ақпаратты өзгертетін немесе өшіретіндей ақпаратқа ие деген сөз, сонымен қоса, белгілі бір қолданушыда сондай мүмкіндік болу қажет.

Өздігінен документтенетін хаттар. Кез-келген хат өзін орындауға қажетті ақпаратқа ие болады. Мысалы, шақырылатын парсер Internet media type-тің көмегімен бейнелене алады. Жауаптар, олардың кәштеу мүмкіндігі бар екендігін дәлелдейді.

Клиенттер ақпаратты body денесі арқылы, сұраныс жолының параметрлері арқылы, тақырыбтың сұранысы арқылы немесе сұратылған URI арқылы. Бұл гипермедиа деп аталады. Жоғарыда келтірілген мысалдай, Бірегей интерфейсстің тағы бір мағынасы, кез-келген REST сервис веб-беттің іргетастық дизайнын жоғалтпау керек.

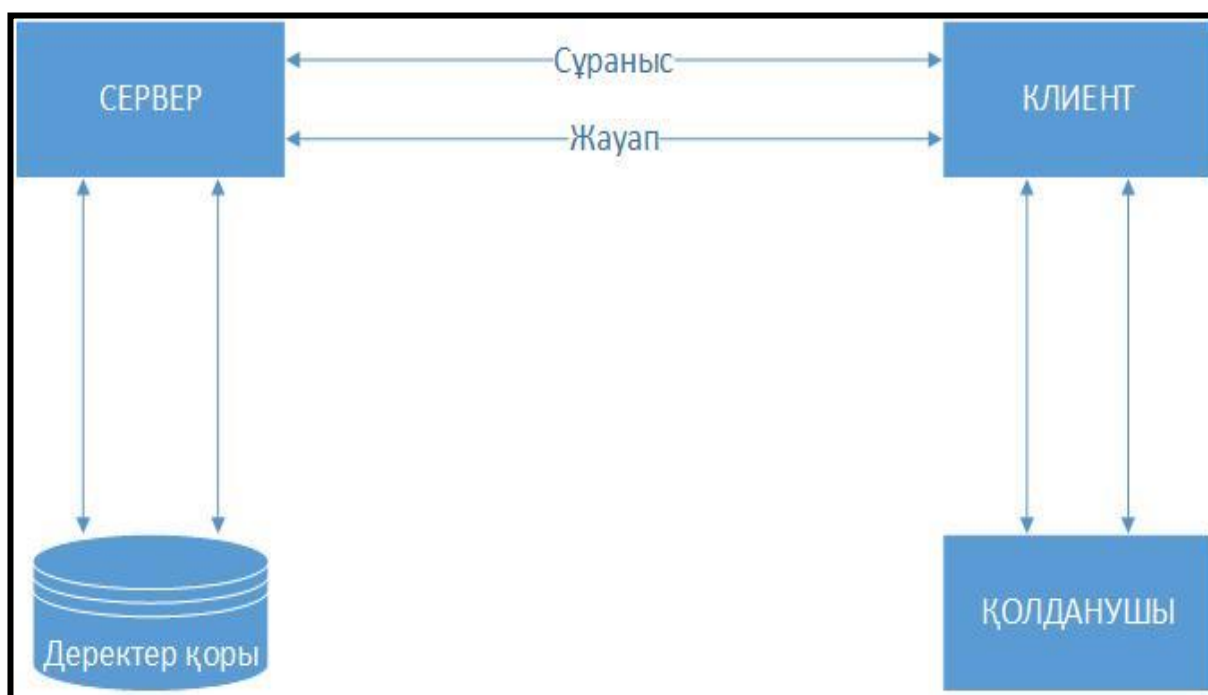
Күйдің болмауы (Stateless). REST Representational State Transfer-ге акроним болғандықтан, күйдің жоқ болуы маңызды сәттердің бірі. Осыған байланысты, сұранысты өңдейтін күйдің сұраныста орналасады, немесе, URI-дің ішінде, немесе, сұраныс жолағының параметрлерінде орналасады. URI уникалды түрде ресурсты жекешелендіреді, және дене осы ресурстың күйін сақтайды. Сервер өндеуді аяқтағаннан кейін, күй толық түрде немесе бөлінген түрде клиентке жауаптың тақырыбы, статусы және жауап денесінде жіберіледі.

Бағдарламалаудағы жүрген адамдардың көбісі, контейнердегі бағдарламалауға, яғни, бірнеше сұраныстардың күйлерін сақтайтын сессиямен жұмыс жасауға үйреніп қалған. REST технологиясында, сұранысты орындату



үшін серверге арналған барлық ақпаратты қосу керек, ал күйді клиент қажеттілік бойынша, және егер осы күй бірнеше сұранысты қамтитын болса ғана қосады. Күйдің болмауы деректерді барынша қамтуды қамтамасыздандырады, және сервер сеанстың күйі арқылы қатынаспайды. Бұдан басқа, теңдікті бақылаушыға сессия мен жүйенің байланысы туралы уайымдамауға болады.

Қолданушы-сервер бірегей интерфейс клиенттерді серверден алыстатады. Интерфейстерді ажырату әрбір сервердің ішіндегі және деректер сақтамайтын типтегі клиенттер болады. Қолданушының интерфейсімен немесе қолданушының күйімен байланыспаған, сондықтан, серверлер жеңілдейді және тез өзгеріске ұшатын мүмкіндікке ие болады. Сервер мен клиент, интерфейс өзгергенше жеке ауыстырыла алады және жеке дамыту мүмкіншілігіне ие болады. Қолданушы-сервер технологиясы 1.2-суретінде көрсетілген.



**1.2-сурет – Қолданушы-сервер технологиясы**

Күймен ресурстың арасындағы айырмашылықты анықтап алу керек. Күй немесе қосымшаның күйі деген, қазіргі сәттегі сұранысты немесе сессияны қанағаттандыратын, деректерді алуды қамтамасыздандырады. Ресурстық күй немесе ресурс, бұл ресурстың көрінісін анықтайтын деректер көзі. Мысалға: деректер қорында сақталған мәлімет. Кейін батырмасын басқанда, әр уақытта алатын нәтиженің айырмашылығы күйдің жоқтығымен байланысты болып табылады. Жауапты кәштеу(Cacheable). World Wide Web сияқты, клиентте жауаптарды кәштей алады. Жауаптар өздерін нақты түрде немесе нақты емес түрде кәштеленетін немесе кәштелмейтін етіп, көрсеткеннің арқасында клиент ескірген немесе қате мәліметтерді келесі сұраныстарда жауап ретінде алмайды. Жақсы құрылған кәштеу толықтай немесе жартылай клиент-серверлік қарым-

қатынасты шегереді, осының көмегі келесідегі жүргізілетін жұмысқа пайдасын тигізеді. Көпдеңгейлі жүйе (Layered System). Қарапайым жағдайда клиент сервермен тікелей байланысқанын немесе басқа біреу арқылы байланысқанын айта алмайды. Өткінші сервер жүйенің тез өзгергіштігін қамтамасыздандыра алады, және соның арқасында түскен қысымның теңдігін қамтамасыздандыра алады, және жалпылама кэш ұсына алады.

Сұраныс бойынша жазылған код (Code on Demand). Серверлер уақытша клиенттің мүмкіншілігін ескере отырып оның мүмкіншілігін кеңейте алады, немесе мүмкіншілігін өзңдей алады. Мысалы, компиляциядан өткен Java-апплеттер немесе Java-да жазылған клиенттік скриптітер болуы мүмкін.

Жекешелендірген жүйе осы қағидаларды ұстана отырып, RESTful архитектураны іске асыра отырып, мынадай мүмкіндіктерге ие бола алады: өнімділік, кеңейтілімдік, оңайлық, өңделетін мүмкіндік, түсініктілік және тұрақтылық.

## 2 Жобалау бөлімі

### 2.1 Унифицирленген моделдеу тілі

UML – бағдарламалық дамытуға графикалық объект модельдеу сипаттау үшін тілді, бизнес-процесстерді модельдеу, жүйелер инженерлік және ұйымдастырушылық құрылымдарды салыстыру.

Ірі бизнес қосымшаларын іске асыратын және компанияның жұмысын қолдайтын ірі корпоративтік қосымшалар жазба модульдерінің жинағы ғана емес болуы керек. Олар стресстік жағдайлар кезінде ауқымдылықты, қауіпсіздікті және сенімді өнімділікті қамтамасыз ету үшін құрылымдалған болуы керек және олардың құрылымы жиі олардың архитектурасы деп аталады. Сервистік бағдарламашылар «жылдам!» деген секілді сөздер, Пайда болатын қатені табуға және түзетуге авторлар басқа жобаларға ауысқаннан кейін көп уақыт өткен. Яғни, бұл бағдарламалар көптеген салаларда жақсы жұмыс істеу үшін жасалуы тиіс, ал бизнес-функционалдылық – бұл жалғыз емес (бұл, сөзсіз, негізгі ядро). Әрине, жақсы жобаланған архитектура кез-келген бағдарламаға пайдалы, бұл жерде айтылғандай, ең үлкен емес. Біз біріншіден ірі қосымшаларды атап өттік, себебі құрылым - бұл күрделілікпен күресу тәсілі, сондықтан құрылымның артықшылықтары (және біз көрсеткендей, модельдеу және дизайн) қолданба мөлшерінің артуымен біріктіріледі. Құрылымның тағы бір артықшылығы – бұл кодты қайта пайдалану мүмкіндігін береді. Дизайн уақыты – бұл бағдарламаны оқшау модульдер немесе компоненттер жиынтығы ретінде құрудың ең қарапайым уақыты. Кәсіпорындар компоненттік үлгілердің кітапханасын жасайды, олардың әрқайсысы кодтық модульдер кітапханасында сақталған іске асыруды білдіреді. Басқа қолданбаға бірдей функционалдылық қажет болғанда, дизайнер өзінің модулін кітапханадан тез импорттай алады. Кодтау кезінде, әзірлеуші код модулін қолданбаға жылдам импорттай алады.

Модельдеу – кодтау алдында бағдарламалық жасақтама қосымшаларын әзірлеу. Модельдеу үлкен бағдарламалық жасақтамалардың ажырамас бөлігі болып табылады және орта және тіпті кішігірім жобаларға да пайдалы. Модель аспан құрылысы кезінде сызбалар мен басқа жоспарлар (торап картасы, қасбеттер, физикалық модельдер) ойнайтын бағдарламалық жасақтаманың дамуында ұқсас рөл атқарады. Модельді пайдалана отырып, бағдарламалық қамтамасыз етуді әзірлеу жобасының табысты болуына жауапты тұлғалар бизнес-функционалдылықтың толық және дұрыс екендігін қамтамасыз етуі мүмкін, түпкі пайдаланушының қажеттіліктері қанағаттандырылады және бағдарламаның құрылымы іске асырылмай тұрып ауқымдылық, сенімділік, қауіпсіздік, кеңейтілім және басқа да сипаттамаларға қойылатын талаптарды қолдайды. кодты өзгерту қиын және қымбат. Зерттеулер көрсеткендей, ірі бағдарламалық жасақтамалардың сәтсіздікке ұшырауы үлкен мүмкіндікке ие - шын мәнінде, үлкен бағдарламалық жасақтама қосымшасы өзінің барлық талаптарын уақытында және бюджет шеңберінде орындауға қабілетті емес, бұл

сәтті болады. Егер сіз осы жобалардың бірін жасап жатсаңыз, табысқа жету мүмкіндігін арттыру үшін бар мүмкіндігіңізді жасауыңыз қажет және модельдеу – сіздің командаңызды кодтауды бастамас бұрын, сіздің дизайнныңызды визуализациялаудың және оны талаптарды тексерудің жалғыз жолы.

## 2.2 Прецеденттер диаграммасын құру

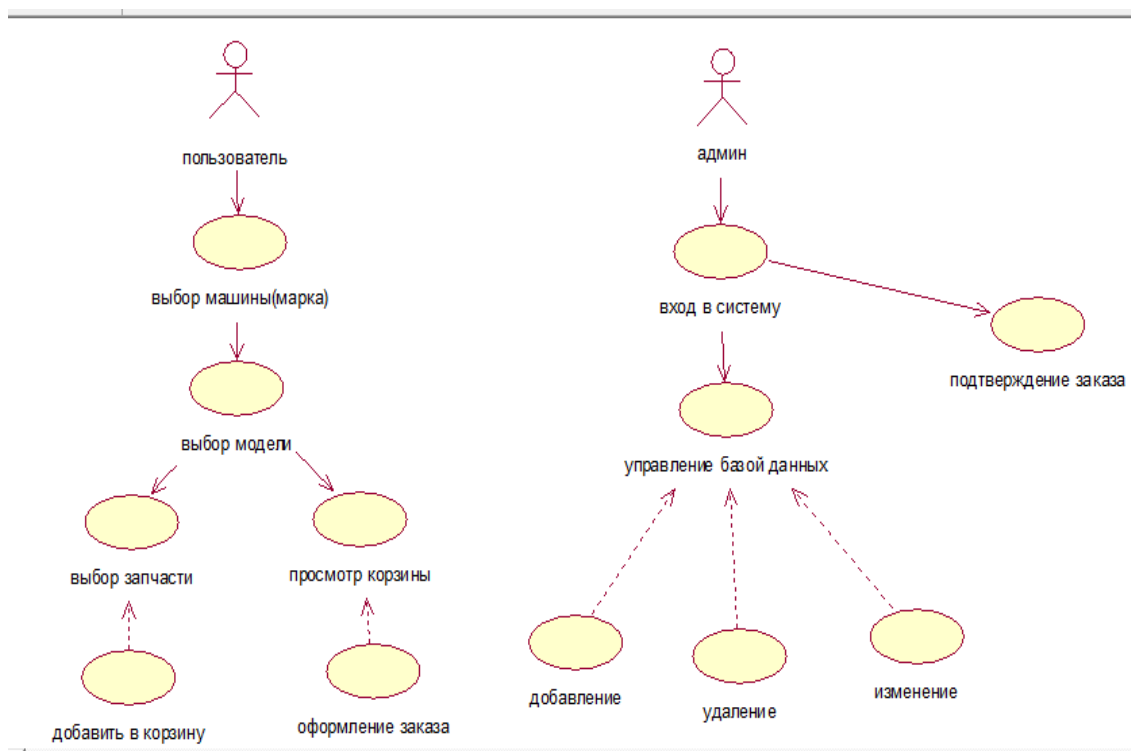
Прецеденттер диаграммасы кез-келген проектіні жобалауда оны қолданушылар ретінде көруге мүмкіндік береді. «AutoShop» – мобильді қосымшасында қолданушылардың 2 түрі бар. Use Case диаграммасынан көріп отырғанымыздай әр қолданушының өзіне тән іс-әрекеті бар. Қолданушының бірінші типі – «сатып алушы» жүйедегі мүмкіндіктері керекті көлік құрылғысын таңдау және қажет тапсырыс рәсімдеу. «Администратор» типіндегі қолданушыға тоқталатын болсақ, оның жүйедегі мүмкіндіктері «сатып алушыға» қарағанда көп. Бұл қолданушы жүйеге өзгерістер енгізе алады. Сонымен қатар, көлік құрал-жабдықтарын деректер қорына енгізу, деректер қорынан жою және құралжабдық мінездемесіне өзгерістер енгізе алады. Сатып алушы рәсімдеген тапсырысты растау іс-әрекеті қарастырылған. Жүйені құру үшін арналған зат обласын толықтай білу қажет. Сонымен қоса жүйенің қалай жұмыс жасайтыны туралы ақпараттар болу қажет. Диаграмма жүйесі ретінде орындайтын операция тізімдерін құру үшін тағайындалған. Ол берілген методтағы жүйе объектісі болып анықталған тізім жүйесін табжылмай орындайды. Осындай түрде жасалатын жүйенің функция құрылымы құрылады, сонымен қоса дүйемен байланысқан объектілердің барлығының іс-әрекетінің жоспарлары жазылады. Прецедент диаграммасы процесін модельдеу үшін пайдаланушы іс-әрекетін толықтай көрсетуіне, сонымен қоса толық реакциясына да бағытталған. Use Case диаграммасы төменгі суреттерде көрсетілген.

Жүйенің жасалатын жұмысы онымен қалай айналысатынына және не қажет болуына байланысты болады. Объект моделі өзгергіштік тараудағы прецедент моделімен байланысты болады. Динамикалық модельді құру негізінде «сырттан ішке» деп аталады. Орналасқан жүйеден басқа жерде болуы мүмкін пайдаланушы зерттеуінен, сонымен қоса ақпараттың қимыл бөліктерін ашу жолынан басталады. Осындай тұрыптан ақпарат құрылымы пайда болады, яғни, қажет ететін іс-әрекеттің ұсынуынан болады. Ішкі бағыттың жылғуы, сонымен қоса оның қадамы бөлек бір жоспары болып табылады.

Қолдану типтері (Use Case) жүйе және активті субъект арасында болатын диалогты модельдеуге мүмкіндік ұсынады, сонымен қоса функцияны соңына қарай бейнелейді. Жүйені қолдану типтер жиыны оны пайдаланудың көптеген амалдар назарына бағытталған. Қолдану типтері – транзакция жүйесі бойынша орындалатын тізбектілік, осы арқылы табылған активті субъект, яғни, керекті

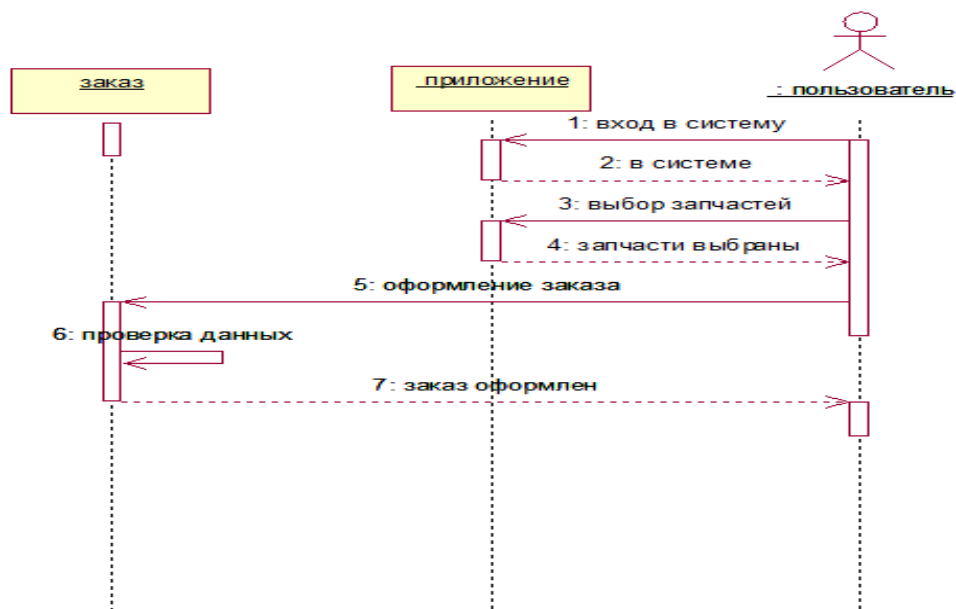


нәтижені оңай алуға болады. 2.1-суретте қолданушының прецеденттер диаграммасы көрсетілген.



**2.1-сурет – Қолданушының прецеденттер диаграммасы**

Қолдану диаграммасы (Use Case Diagram) – бұл активті денелердің көптеген графикалық көрінісі оны қолданудың, ол болмаса басқа нұсқалардың амалдарымен өзара байланысады. Жүйені жоспарлағанда құрылымның кілттік функциясын, сонымен қатар көптеген қолданушылар ұсынатын негізгі диаграмма (Main Use Case Diagram) осылай конструкцияланады. Жүйені құру үшін арналған зат обласын толықтай білу қажет. Сонымен қоса жүйенің қалай жұмыс жасайтыны туралы ақпараттар болу қажет. Диаграмма жүйесі ретінде орындайтын операция тізімдерін құру үшін тағайындалған. Ол берілген методтағы жүйе объектісі болып анықталған тізім жүйесін табылмай орындайды. 2.2-суретінде Администратор прецеденттер диаграммасы көрсетілген.



**2.2-сурет – Администратор прецеденттер диаграммасы**

Use Case – бұл белгілі актерге (Actor) арналған, белгіленген бір нәтиже бере алатындай, бағдарлама жүйесімен орындалатын іс-қимылдар тізбегінің сипаттамасы. Ал, прецедент негізінен моделдегі берілген мәндердің тәртіптік структурасын жасайды, сонымен қатар прецеденттер кооперацияның қатысуымен таратылады. Актер (Actor) – бұл берілген жүйедегі негізгі деген элементтермен байланыста болатын негізгі объектінің рөлі болып саналады. Актер мен қолданушының салыстырмалы түрде айырмашылығы келесідей: қолданушы жүйені толықтай пайдаланатын шынайы объект. Ол бірнеше рөлде ойнауы мүмкін болады, соған байланысты ол бір емес, бірнеше актер болуы мүмкін.

### 2.3 Тізбек диаграммасын құру

Тізбек диаграммасы уақыт бойынша берілген объектілердің өзара ұйымдасу операцияларының жасалу ретін безендіреді және жоспарлармен қарастырылған мүмкіндіктердің орындалу процесіндегі өзара алмасатын объектілердің хабарлама тізбектерінің реттелген класстар және объектілер бейнеленеді. Тізбектер диаграммасы (Sequences) – қолданушының іс-әрекетін уақытқа қатысты қарастырады. Яғни администратор қолданушысы жүйеге кірмей мобильді қосымшаға өзгерістер енгізе алмайды, деректер қорымен жұмыс істей алмайды, тапсырысты растай алмайды. Сол сияқты сатып алушы керекті жабдықты таңдамас бұрын тапсырысты рәсімдей алмайды. Бұл дегеніміз сатып алушы ең алдымен керек көлік жабдығын таңдап оны «корзинаға» орналастыру жүзеге асқаннан кейін тапсырыс беру іс-әрекеті орындалады.

Тізбектелген диаграммалар жалпы түрде Logical View пакетіндегі

көзделген қолдану нұсқаларының жүзеге асырылуымен бірге белгіленеді. Тізбек диаграмма негізгі төрт элементтен тұрады:

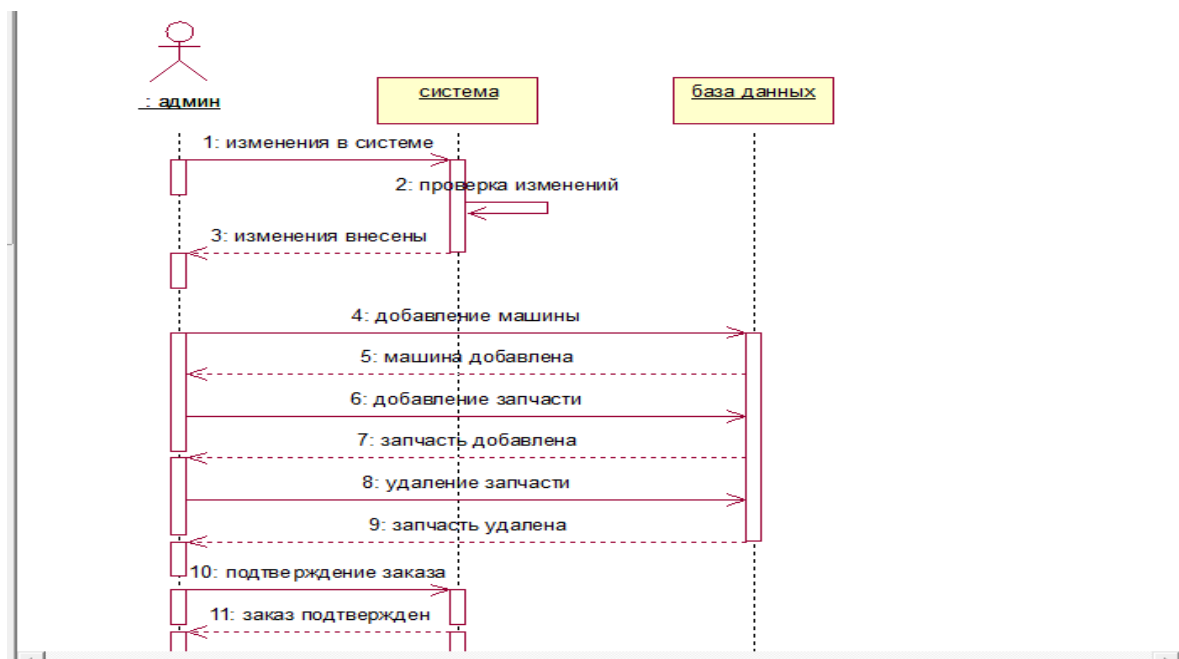
– прецеденттегі артындағы мәтінінің іс-қимылы. Ол жоғарыдан төменге жазылады және сол жақтан басталады. Сол терезеде іс-қимыл сипатталуы болып, орындалатын жұмыс уақытындағы белгілі ақпараттар қызмет етеді;

– объектілер "объект-класс" кейпінде аты болмаса объект данасының нөмірі және класс объектісінің атауы жазылады;

– хабарландыру, бір бағытта көрсетілген белгілі бір объектіден келесі объектіге бағытталған іс-қимыл туралы мәлімет хатынан тұрады. Берілген бір уақытта орындалуы мүмкін және осы іс-әрекеттегі берілген жүйенің қайтарымды реакциясы болуы мүмкін болып келеді;

– әдістері (операциялар). Тікбұрыш түрінде болады. Олар үздік сызықпен бейнеленген. Яғни, берілген әдістерге кіретін сол объектілер болып табылады. Тік бұрыштың ұзындығы артынан қуушыда басқару нүутесін көрсетуде қолданса болады: Тікбұрыш аяқталатын немесе бітетін әдіс толықтай нүктесіне дейін басқарумен игеріледі. Бұл, берілген үшбұрыштар циклге арналған объекті түзуі деп саналады. UML талаптарына байланысты объектінің тізбектелген диаграммасында тіктөртбұрыш кейпінде көрсетіледі. Объектіні 3 түрлі есіммен атауға болады: тек оның атауын көрсетіп кету қажет, объект пен класстың атауын беру қажет, не класстың атауымен шектелу керек.

2.3-суретінде тізбектер диаграммасы төменде көрсетілген.



2.3-сурет – Тізбек диаграммасы

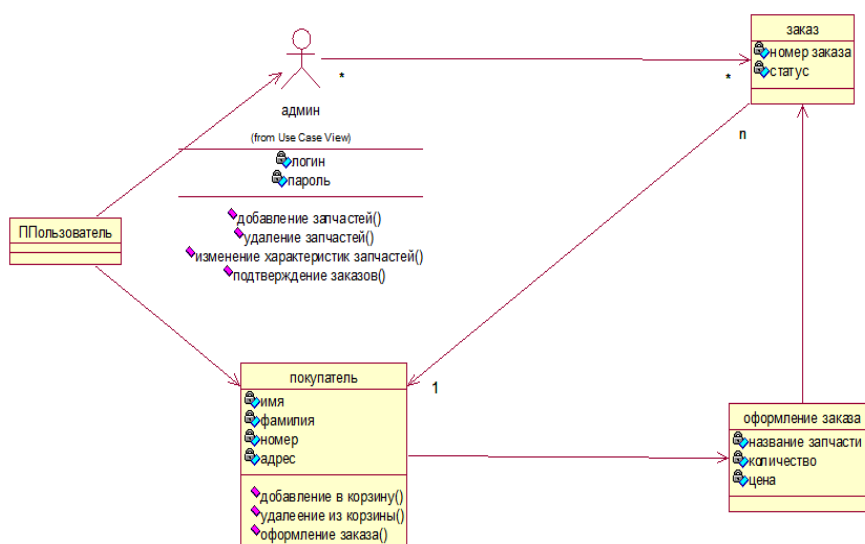
Тізбектелген диаграммада активті объектілермен өзара бірігу мәліметтер жүйесінің көрсетілуін рұқсат ететін шекүтеулер класы қосылады.

## 2.4 Күй диаграммасын құру

Күй (State) – бұл кейбір нышандарды орындау кезінде белгіленген әрекет болмаса басқа оқиғаның келуін күтуде мекендеуді жүзеге асыратын жиынның шарты. Объектінің күйі класы бірнеше немесе бір деректермен бейнеленеді.

Күй диаграммасы – нысанның бір қалпынан басқасына ауысуын айыратын, және іс-әрекеттері қалпыларын алмастырумен ескертілінген, оқиғалардың, болмаса хабарлаулардың объекттік жағдайларының графикалық түрдегі ұсынысы. Сондай-ақ, күйлер диаграммасы объектінің бір күйден екінші күйге өтуін суреттеу үшін қолданылады. Диаграммалардың бұл типінің басты элементтері күй және өту болып табылады. «AutoShop» мобильді-қосымшасында тапсырыс берудің күйлерін қарастыратын болсақ, ең алдымен, тапсырыс расталу күйіне өтеді. Келесі күйі – тапсырысты тексеру. Тексеру кезінде қателіктер болмаған жағдайда тапсырыс расталады. Объектінің өз қалпын уақыт бойында қай бағытта өзгертетінін түсіну үшін жағдайдың өзгеруінің спецификациясы бар. Субъектінің жағдайы оның атрибуттарының, және қатынас атрибуттарының мағынасымен сипатталады.

Күй спецификациясы класс атрибутын сипаттайды. Тәртіп спецификациясы класс оталарын анықтайды, олардың кейбірі нысан күйін басқаға өзгерте алады. Объектілер жағдайларын моделдеу күйлер диаграммасының арқасында жүзеге асады. Күйлер моделі системаға маңызды кластар үшін сызылады. Объектілер белгілі бір нүктеден шығып, соңғы нүктеге жеткенше барлық нүктелерден өтеу және соңғы нүктеге жетеді. 2.4-суретінде күй диаграммасы суретте көрсетілген.



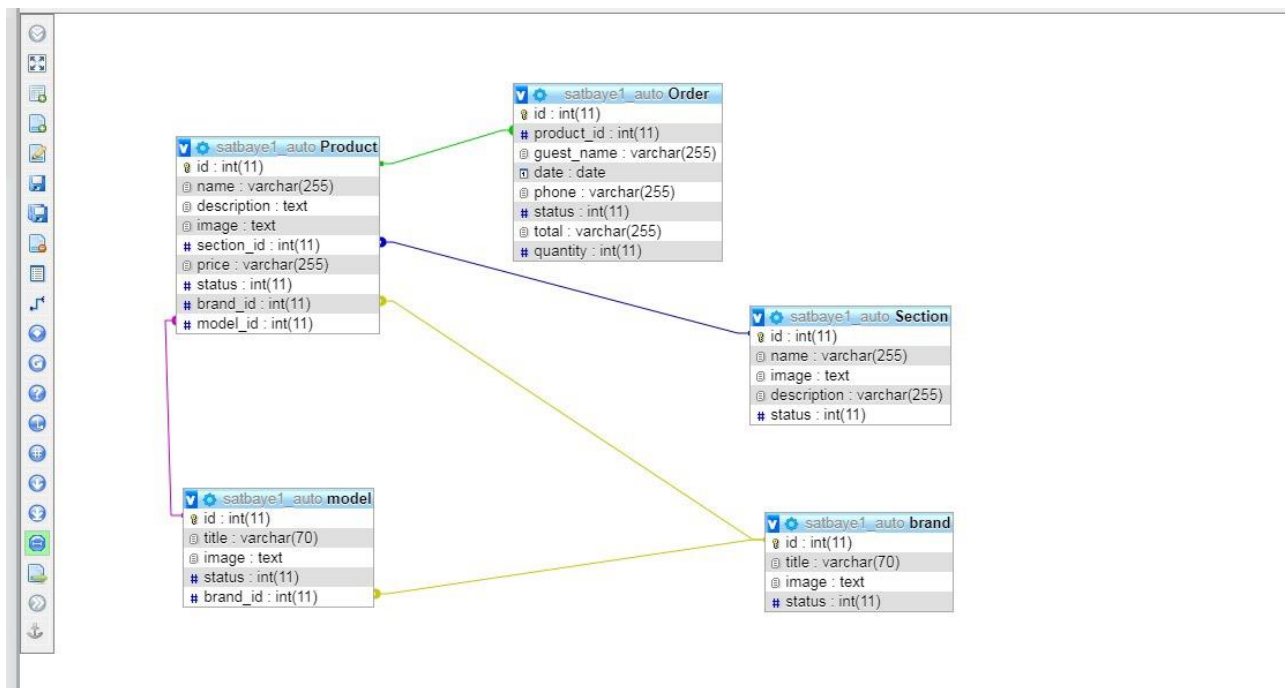
2.4-сурет – Күй диаграммасы

## 2.5 ER диаграммасын құру

Кәсіпорынның өзара қарым-қатынасы диаграммасы (ER) – адамдар, нысандар немесе тұжырымдар сияқты «субъектілер» жүйеде бір-бірімен байланысты болғандығын көрсететін блок-схарттың түрі. ER диаграммалары көбінесе бағдарламалық жасақтама жасау, бизнеске, білімге және зерттеуге арналған ақпараттық жүйелерде реляциялық деректер қорын жобалауға немесе жөндеуге қолданылады. Сондай-ақ ERD немесе ER модельдері ретінде белгілі, олар субъектілердің қарым-қатынасын, қарым-қатынастарын және олардың атрибуттарын бейнелеу үшін төртбұрыш, ромбус, овал және байланыс желісі сияқты таңбалар жиынтығын пайдаланады. Олар грамматикалық құрылымды, зат есімдермен және етістік қатынастарымен ерекшеленеді.

ER диаграммалары нысандардың өздері арасындағы қарым-қатынасқа емес, нысандардағы элементтердің қарым-қатынасына шоғырланған деректер құрылымының диаграммаларымен (DSD) байланысты. ER диаграммалары жиі үдерістер мен жүйелер үшін ақпараттар ағыны көрсетілетін деректер ағынын диаграммасымен (DFD) бірге жиі пайдаланылады. ER диаграммалары логикалық және бизнес ережелерімен (логикалық деректер үлгісінде) және нақты физикалық модельдерде (физикалық деректер моделінде) реляциялық дерекқорларды модельдеу және жобалау үшін пайдаланылады. Бағдарламалық жасақтаманы әзірлеуде ER графикасы көбінесе ақпараттық жүйе жобасына қойылатын талаптарды анықтаудағы алғашқы қадам болып табылады. Кейінірек ол сондай-ақ нақты дерекқорды немесе деректер қорын модельдеу үшін қолданылады. Реляциялық деректер базасы баламалы реляциялық кестеге ие және қажет болған жағдайда осылайша көрсетілуі мүмкін. Егер деректер әртүрлі өрістерге, жолдарға немесе бағандарға нақты бөлінбесе, ER диаграммалары шектеулі қолданысқа ие болады. Сол жартылай құрылымдалған деректерге де қатысты, өйткені кейбір деректер тек пайдалы болады.

ER-диаграммалар бизнес-процесстерді қайта құруда қолданылатын дерекқорларды талдауға көмектеседі және жаңа дерекқор теңшелімін модельдеуге көмектеседі. ER диаграммасы 2.5-суретте көрсетілген.



2.5-сурет – ER диаграммасы



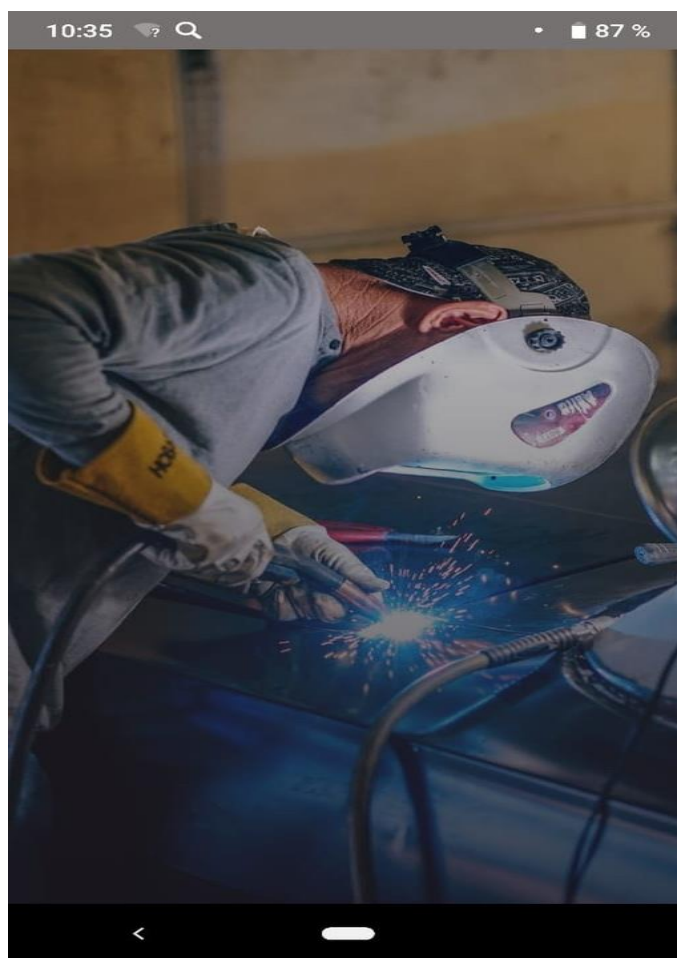
## 3 Құрылым бөлім

### 3.1 Қолданушы интерфейсі

Бұл проектіні жасаудағы түпкі мақсаты желіде орналастыру және мобильді қосымшаға REST технологиясы арқылы дерек қорын беру болып табылады. Мобильді қосымшаны кез-келген Android телефоны бар адам қолдана алады. Мобильді қосымшаның тоқтамай жұмыс жасап тұру үшін мобильді қосымшаның толықтай серверде орналасып, желіге тікелей хостинг тұрғысы арқылы таратылуы қажет.

Мобильді қосымша ретінде Apache 2.0 сервері таңданып алынды. Хостингтер тізімінен [www.auto.vtipo1519.kz](http://www.auto.vtipo1519.kz) хостингі таңданып алынды.

Мобильді қосымшаға кірген бойда беткі фондық суреті тұрады. Екі кезекті қойындысы автокөлік түрілерімен ашылады. Кезекті қойындылардың таңдап қалағанына кірсе болады, оң жақ үстінгі бөлігінде сауда-саттық қоржынын көре аласыз. Мобильді қосымшаның басты беті 3.1-суретінде көрсетілген.



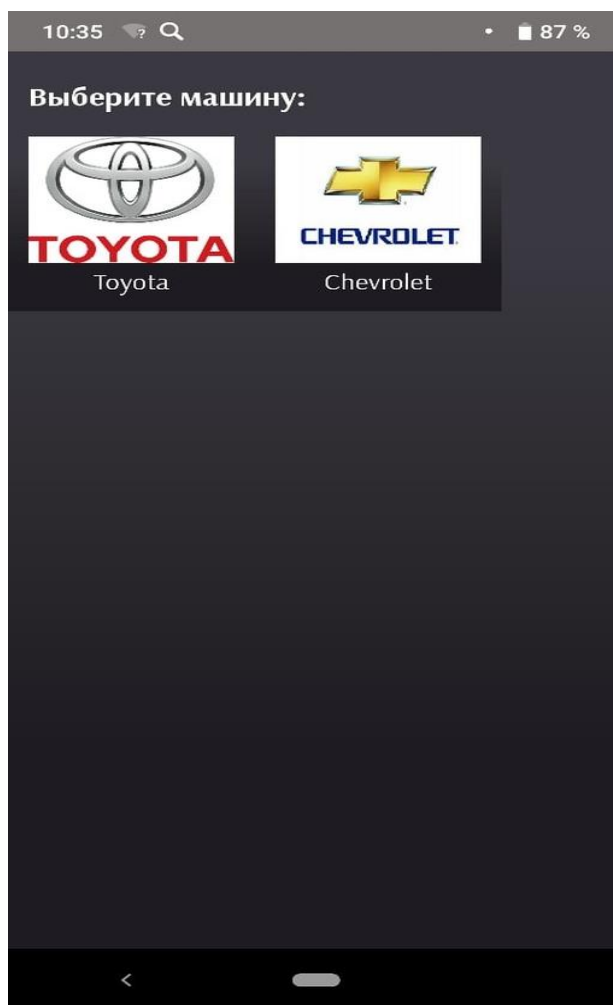
3.1-сурет – Мобильді қосымшаның басты беті

Мобильді қосымшаның бастапқы беті пайдаланғандықтан қандай да құрылғымен ашса да, ол адам көзіне жағымды болып көрінеді.

Ұялы телефоннан, планшеттен немесе дербес компьютермен ашса да веб-беттің ешқандай элементі назардан жоғалып кетпейді.

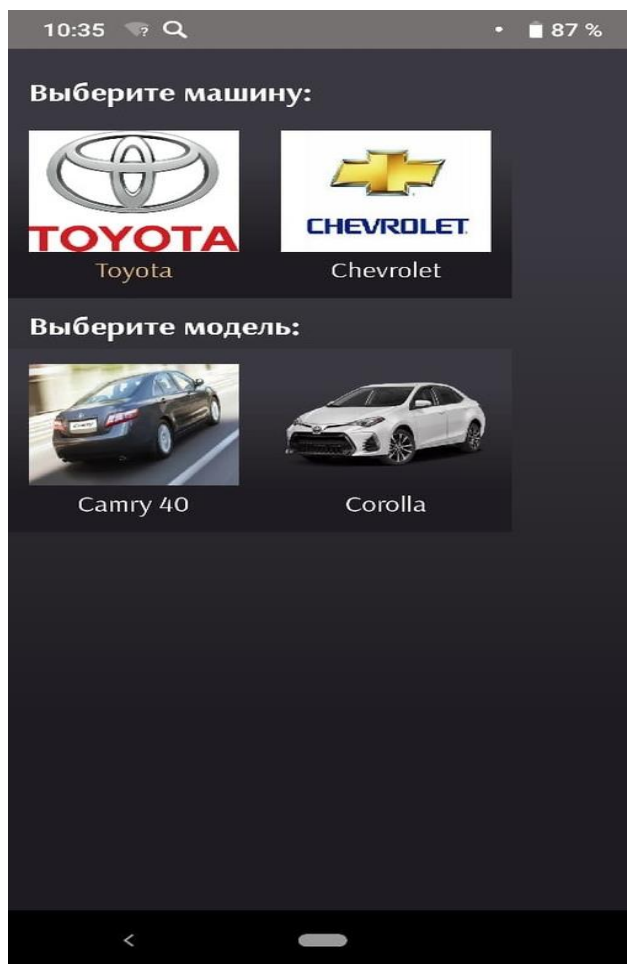
Жоғарыда тұрған бастапқы беті тірегіне сай бір жылжымалы блок сияқты бір элементтің астына жасырынып тұрады. Сол батырманы басқанда, мәзір элементтері астыға жылжып, барлығы көрінеді. Мобильді қосымшаның ұялы телефоннан кіргендегі қалпы суретте көрсетілген. Кез-келген құрылғыдан ашқанда адаптивті дизайн өзінің жұмысын жасап, сол экранға бейімдеп береді.

Мысалы, планшетпен мобильді қосымшаға кірсе адаптивті емес тірекпен жасағандықтан экран бетіне толығымен үлкен болып тұрады.



### 3.2-сурет – Мобильді қосымшаның ұялы телефоннан кіргендегі қалпы

Мобильді қосымшаның екі бөліктегі машина модельдері бетіне өтіп, қалаған екеуінің біреуіне кіріп керекті бөлшекті издесеңіз болады. Еш тіркелусіз кірсе болады, «Автобөлшек» қосымшасы дүкенді түрде болғандықтан оңай кірсеңіз болады. Мобильді қосымшаның таңдаулы парақшасы көрсетілген.



**3.3-сурет – Қосымшаның таңдаулы парақшасы**

Қосымшаның екі түрлі көлік таңдау парақшасы көрсетілген әрі моделі сурет ретінде көрсетілген. Келешекте дами отыра тағы бөлшек модельдірі қосылуда болады. Көлік моделін таңдаған кезде төменгі бөлігінде басуға ыңғайлы жерде жалғастыру бөлігі шығады.

Келешекте проекттің дамуына қарай көліктерді қосуға болады, сол елдің моделінен бірнеше көліктерді таңдауға болады. Автомобиль жасау саласындағы жетекші әзірлеушілер халықаралық стандарт бойынша сапалықты пайдаланады. Бұл бірнеше маман қатысатын, олардың әрқайсысы ішкі толтыру, бөлшектер, корпус, дизайн саласына жауап беретін қажырлы еңбек. Автомобиль концернінің жөнге салынған жұмысы кезінде идеядан бастап макетті құру мен барлық тетіктерді толық өңдеу мен бірінші машинаны конвейерден алу процесі 72 айға жуық өтеді. Түзетулер болуы мүмкін, олар негізінен қандай жаңа технологиялар жасалғанына, бұрынғы модельдің негізіне алынғанына, өндіруші қандай өндірістік қуатты ұсынатынына байланысты.

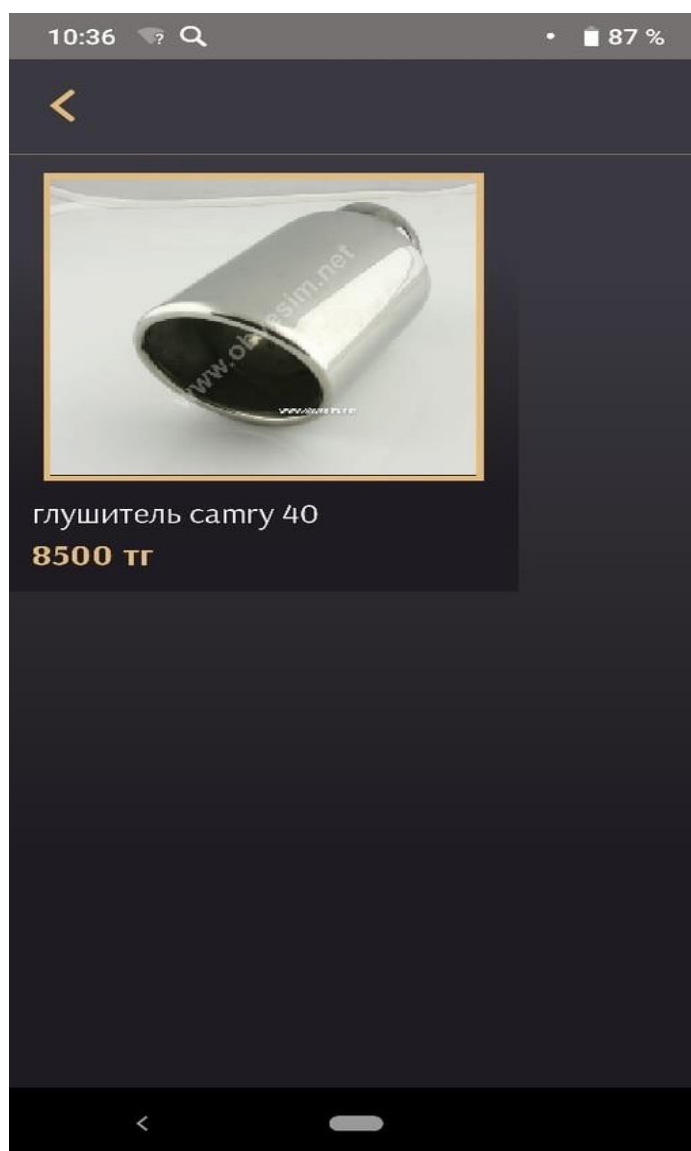


### 3.3-сурет – Қосымшаның Бөлшектер парақшасы

Бөлшектер тізімінде керекті бөлшекті сауда-саттық қоржынын оң жақ үстінгі бөліктін ішінде көрсе болады. Ол бетте таңдаған бөлшек сақталып тұрады. Ішкі құрылымдарын әр бөлшектің ішінде оқыса болады. Үстінгі көлік атауын басып артқа қайтарса болады. Рестайлинг, яғни, қазір бар көлікті қайта жасау, әрине қиындыққа соғады.

Автокөліктегі кондиционердің ең көп тараған мәселесі-хладагенттің ағуы, оларсыз кондиционер көшеден түсетін жылы ауаны салқындатуы мүмкін емес. Сондай-ақ, қазіргі заманғы автомобильдердің иелері кездесетін өте жиі проблема-кондиционер компрессорының қоректенуі. Көптеген автокөлік жабдықтары сияқты компрессор көлік құралының электр желісінен қоректенеді.

Электрикамен проблемалар болған жағдайда кондиционер компрессоры электр энергиясының жеткіліксіз мөлшерін ала алады. Нәтижесінде ол ауаны салқындата алмайды. Кейбір компрессорлар кондиционер бар өз конструкциясында вал, ол тозуы мүмкін.



### 3.4-сурет – Сауда-саттық қоржынының парақшасы

Сауда-саттық қоржынында қолданушы керекті бөлшегін сақтап кез келген уақытта тапсырыс жасап кейін, қолданушы тапсырысшымен хабарласып, тапсырысын келіскен уақытта, керекті жеріне апарып береді.

Тапсырыс жасау парақшасы төменде көрсетілген.

Бөлшектердің сынуының жиі себебі компонент щеткаларының тозуы немесе генератор подшипниктерінің істен шығуы болып табылады. Бұл жағдайда проблеманы шешудің екі түрі бар-жаңа генераторды сатып алу немесе ескі жөндеу. Дегенмен, ескі генератордың қалқаны әрдайым мүмкін емес және аз сенімді.



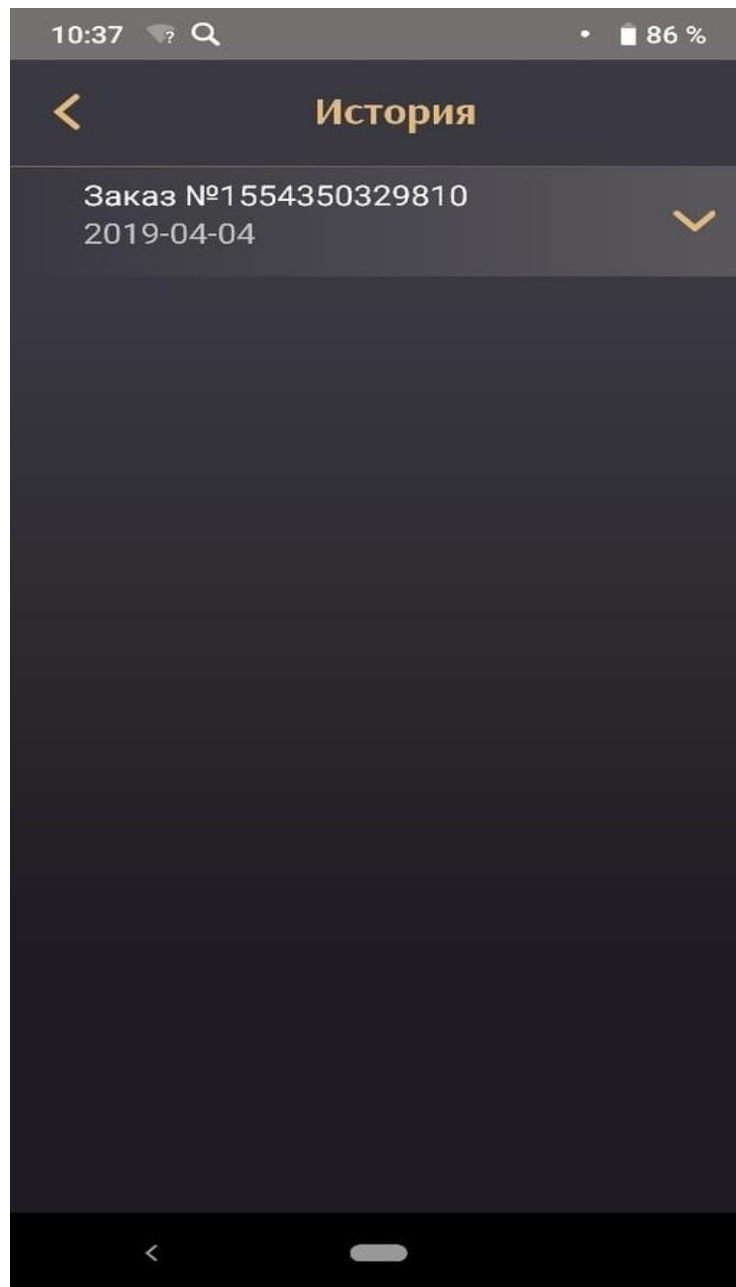
### 3.5-сурет – Тапсырыс жасау парақшасы

Қолданушы бетіне өтіп, өзінің жеке ақпараттарын енгізіп тапсырыс жібере алады. Жасалған тапсырыстар тарихтық жерде сақталады, яғни қанша тапсырыс жасалғаны жөнінде көре алады. Төменде тарихтық парақшасы көрсетілген.

Өкінішке орай, көлік құралдарының көптеген иелері көлік құралымен таныс емес, оған қызмет көрсету және жөндеу туралы айтпағанда. Иә, техникалық дилерлік орталықтарда автокөліктерге қызмет көрсету әлдеқайда ыңғайлы, онда сізге қосалқы бөлшектерді өз бетінше сатып алудың қажеті жоқ.

Бірақ біз әрбір жүргізуші көлік құралының неден тұрғанын, қандай да бір компоненттер қалай жұмыс істейтінін білуі тиіс деп есептейміз. Осының арқасында сіздің әрқайсыңыз дилерге төлемей, қосалқы бөлшектерді өз бетінше сатып ала алады.





**3.6-сурет – Тапсырыстың сақталуы**

Әр тапсырыстың күні мен номерлері, қанша және қай күні жасалғанын көрсете болады. Төменгі бөлікте тапсырыстың нешеуі керектігі көрсетілген. Тізім ретінде толып отыра береді керек кезде өшіріп алып тастаса болады.

Бұл бөлігі әр адамның тарихы ретінде қанша уақыт бойы не алғанын қарап сақтап отыра алады және керек емес болған жағдайда өшіріпте тастай алады. Кейде тапсырыс тарихынан бөлшекте бірақ болған жағдайда қайтарып алса болады, яғни чектің орнын басқара отырады.



**3.7-сурет – Тапсырыс неше керекті панелдері**

Тапсырыстың санын көбейтіп алса болады Әр тапсырыста тізім арқылы көрсетіліп отырады. Сақталған қордың ішінен шықпай өзгертсе болады. Әр тапсырысты көбейткен сайын автоматты түрде бағасы көбейіп көрсетіле отырады. Ол тапсырыстар тікелей тарихта чек ретінде сақталып отырады. Әр телефонда скрин функциясы бар сол арқылы сақтап, тарихты тазартып отырса болады.

Қолданушылар тапсырыс бергеннен кейін, тікелей администраторға тапсырыс келеді. Келешекте дами отыра бұл заказ тек белгілі қызметкер қабылдап отырады, әлбетте қосымшаның дамуына қарай байланысты болады. Тапсырыстан кейін қолданушы өз ақпараттарын және жеке номерін еңгізіп тапсырысын ала алады.

## ҚОРЫТЫНДЫ

Бұл проектiнiң жасалу барысында қолданушыға деген барлық қолайлылық көзделуде. Басқа да ұқсас проектiлерге қарағанда артықшылықтарды арттыру мақсаттың бiрi.

Проектiнiң қызметi мобильдiк қосымшаның тұрақты және қолданушыларға ыңғайлы болып табылады. Мобильдi қосымша үлкен аудиторияға арналған (кез-келген адамға), сондықтан мобильдi қосымша жоғарғы технологияларға сәйкес болуы керек, тартымды дизайнға, және ыңғайлы интерфейске ие болуы керек және де, көпфункционалды, кез-келген қосымша қолданушыға ыңғайлы болуы керек.

Жоба бойынша, Android мобильдi қосымшасы клиент ретiнде, iшкi жобасын өзгерту қалпын iске асырады. Ең бастысы, қосымшаның кодында жазылған REST шарттарын орындауы қажет болады. Сол кезде, қосымшадағы барлық ақпарат қол жетiмдi болады. Мысалы, белгiлi бiр бағдарлама құрушы, сайтқа арнап мобильдi клиент немесе веб-бет жасағысы келсе, жобаның администраторымен хабарласып, REST сервистiң параметрлерiн қарастырып клиент құра бере алады.

Жалпы бұл қосымшаның нәтижесi әзiргi уақытта керектi бастапқы орындалуына жеттi, алдағы уақыттарда үлкен жоба ретiнде қарастырылады. Оған жоба бойынша топ арқылы проектiнiң толықтай Play Market-пен жарыққа шығуына iс-әрекет ету басты мақсаты болып келедi. Әзiргi бастамасы нәтижеге келдi. Оның қолданушылар алдында мiндетi тапсырыс жiберуден бөлек тиiстi орнына жеткiзу болады және бөлшектердiң жаңа түрлерi әр шыққан белгiсiнде мобильдi қосымшада жаңартылып отырады.

Клиенттермен жұмыс iстеу тәжiрибесi өте маңызды және көбiнесе бизнестi құру және интернеттегi бөлiктердi сату үшiн маңызды. Электронды коммерция шешiмiн таңдағанда, бақытты тұтынушыны есте ұстану керек. Үлгi мобильдi қосымшасын басып, «сатып алуды» жасау арқылы сатып алушының орнына қою керек . Есте сақтай отыра , тұтынушы бөлiгiн тауып, сатып алу үшiн көбiрек қадамдар жасаса, ол сатуға бас тартатын ықтималдығы көбiрек болады. Кейбiр элементтердi басқаларға қарағанда көп немесе аз деп белгiлеу керек – жай ғана мағына ретiнде. Автомобиль қосалқы бөлшектерiн желiде сатқан кезде интернет-дүкендегi бағаны белгiлеуге арналған бiрнеше нұсқа бар. Оны қосымша құнға негiзделген құнға немесе оны тiзiм бойынша бағалай алады. Тиiмдi жеткiзiлiм жүйесi қосымша автокөлiктердi көп кадрларсыз жалдауға көмектеседi. Мүмкiн болса, тапсырысты аяқтап, қадағалау нөмiрiн алуға болады, жапсырманы басып шығарып, клиентке бiр орыннан электронды хат жiбере алады. Электронды коммерция шешiмi арқылы белгiлi бiр деңгейдегi мiндеттеме талап етiледi және басталғаннан кейiн бiрден өкiнуге келмейдi. Бағдарламалық жасақтаманы шешкеннен кейiн, сайтты жұмыс iстеуге және бөлiктерiн сатуға ұзақ уақыт кетпейдi.

Бiз өзiмiздiң қажеттiлiктерiмiзге сай келетiн сенiмдi электрондық

коммерция шешімімен жұмыс істегенде, табысты барынша арттыруға болады. Сонымен қатар, дұрыс платформа таңдау қымбат және сәтсіздікке әкелуі мүмкін.

Сондықтан автомобиль бөлшектерін дұрыс бағытта сату, әрқашан біздің жолымызда.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Максим К., Игров С., Сергей Г. PHP 5: Практика создание Web – сайтов. – СПб.: БХВ – Петербург, 2017. – 150 с.
- 2 Лаура Томсон, Люк Веллинг. Разработка Web-приложений на PHP и MySQL. – СПб.: БХВ – Петербург, 2018. – 365 с.
- 3 Мухаметшин Д.Ф. Описываются приёмы работы с элементами HTML-форм с применением языка программирования PHP. – СПб.: БХВ – Питер, 2015. – 120 с.
- 4 Яргер Р., Риз Дж., Кинг Т. MySQL и MSSQL, Apache. Базы данных для небольших предприятий и Интернета – СПб.: Символ-Плюс, 2016. – 150 с.
- 5 Вагнер Р., Байк А. Энциклопедия JavaScript Киев: Bhv, 2001. – 400 с.
- 6 Дмитриева М.В. JavaScript. Быстрый старт – СПб.: БХВ – Петербург, 2002. – 150 с.
- 7 А. Леоненков. Самоучитель UML (2-е издание). – СПб.: БХВ – Петербург, 2004. – 215 с.
- 8 Скотт Хокинс. Администрирование Web-сервера Apache и руководство по электронной коммерции. – М.: «Вильямс», 2001. – 367 с.
- 9 Ролов А.В., Фролов Г.В. Базы данных в Интернете: практическое руководство по созданию Web-приложений с базами данных «Русская Редакция». – СПб.: БХВ – Петербург, 2000. – 315 с.
- 10 Горев А., Макашарипов С., Владимиров Ю. MySQL и Microsoft SQL Server для профессионалов. – СПб.: Питер, 1998. – 120 с.
- 11 Вагнер Р. Энциклопедия HTML. – Киев: Bhv, 2001. – 400 с.
- 12 Сайтқа кіргенде статьяның аты // Сайттың электронды нұсқасы [https://ru.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://ru.wikipedia.org/wiki/Apache_HTTP_Server)
- 13 Сайт // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/Php>
- 14 Сайт // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/JavaScript>
- 15 Сайт // Сайттың электронды нұсқасы: <https://ru.wikipedia.org/wiki/Yii>
- 16 Ташков П. Веб мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. – СПб.: Питер, 2010. – 512 с. 5 Котеров Д., Костарев А. PHP.
- 17 Ғалымжанова М. Ақпараттық технологияларды сабақта қолдану. – Информатика негіздері, 2006. – 23 б.

## **А қосымшасы (міндетті)**

### **Техникалық тапсырма**

#### **МОДЕЛЬДІҢ ӨМІРЛІК ЦИКЛЫ**

Android платформасындағы «Онлайн Автобөлшек» мобильдік қосымшасының орнату жүйелерін қолдану. Бұл қамтаманы жасау келесі алгоритмдерден тұрады:

- Материалдарды талдау және жинау;
- Техникалық тапсырма құрастыру;
- Мобильдік қосымшасын құру ;
- Қамтама жұмыс жасауы үшін керек болатын барлық программалық компоненттерді автоматты түрде орнату үшін арнайы программаларды орындау;
- Тестілеу;

#### **А.1 Кіріспе**

Андроид платформасы - бұл смартфондарға, планшеттерге және нетбуктарға арналған операциялық жүйе. Google компаниясы бағдарламалық жасақтама әзірлеушісі Android inc болды. Android платформасы арқылы мобильді қосымшасын құрылуын бастау және істеу әдістерін белгілеу;

1. Бағдарламалық қамтаманың орнату, мобильді қосымшасын құрып ,интерфейсін бейнелеу.
2. Хост сатып алу, домен беру.
3. Android платформасындағы Мобильді қосымшасына орнату .
4. REST арқылы деректерді беріп Java қабылдап алу.

#### **А.1.1 Өңдеудің мақсаты мен қызметі**

Бұл бағдарламалық қамтама қолданушылардың терминалдың жұмыс істеу типін анықтап, қолайлы қолдануы арқылы жақсарту мақсатында жасап шығарылады.



Бағдарламаның қолданушылары күнделікті сауалнамаға жауап беру арқылы ыңғайлы қолана алады. Әр қолданушының уақытыг үнемдеуге байланысты кеңестер алады. Күнделікті кеңестерді ала отырып, қандай өзгерістерді еңгізуге болып жатқанын жаза кетеді. Қолданушы жауаптарын ескере отырып кеңестерді жаңалап тұрады.

Қосымшаларды әзірлеудің мақсаты қолданыстағы клиенттердің адалдық дәрежесін арттыру және есептеудің ыңғайлылығы мен қарапайымдылығы есебінен жаңаларын тарту және ПЭК компаниясының бір нүктесінен басқа күшімен жүктерді жеткізуге тапсырыс беру болып табылады.

Ұялы телефондарда жұмыс істегенде экранның бағыты портреттік болып табылады. Верстка экран бағдарының ауысуын ескеруі тиіс: ақпарат блогын ортаға орналастыру, бөрік созу

## **А қосымшасының жалғасы**

### **А.1.2 Қолдану саласы**

Қолдану аймағы – кәмелеттік жасқа толған ұялы телефон қолданушылары.

### **А.1.3 Анықтамалар, терминдер және қысқартулар**

Өндірістік бөлшектер – көпшілік қолданысындағы рецептураны талдау, олардың ассортиментін жаңарту кезеңінде құрамына адам көлігіне зиянсыз, күшті әсерлі бөлшекті зат (жоғары мөлшерде) кірмейтін сенімді бөлшектер өндіру саласы.

## **А.2 Жалпы сипаттамасы**

Мобильді қосымшаның бөлшектері тек қосымшада, қосымшаның келешекте веб-сайтында немесе әлеуметтік желі қосымшалары мен ұялы телефондар арқылы ақшалай төлеуге болады. QIWI Wallet арқылы төлеу ақысыз және комиссиясыз жүзеге асырылады!

QIWI Wallet арқылы RNL-ге қайырымдылық жасау:

Қайырымдылық түрінде, ол RNL веб-сайтында табылған жағдайда, бастапқыда енгізілетін соманы енгізіңіз немесе ұсынылған соманы қалдыруға болады. Содан кейін ұялы телефон нөмірін және қажет болған жағдайда қосымша ақпаратты енгізіңіз. «Төлем» түймешігін басыңыз. Жүйе автоматты түрде төлем үшін шот-фактураны жасайды және сіз оны дереу төлеуге болатын қосымша терезе ашылады, жай ғана QIWI әмияныңыздан құпия сөзді енгізіңіз. Шот-фактураны тез арада төлеу үшін QIWI әмиянынан ұялы телефонның балансынан немесе несие картасының қалдықтарынан төлем алынады.

## **A.2.1 Аппараттық интерфейстер**

Әр қолданушының кез келген төлем қызметтері болады. Бағдарламалық қамтамаға кірген кезде кез келген қызметі төлемі орындалады. Жеке парақшасына кірген уақытта күнделікті керекті төлем ақысы шығады. Әр қолданушы жаңа қызметтері, олар туралы хабар біле алады.

### **A қосымшасының жалғасы**

#### **A.3.1.4 Функционалды талаптарды ұйымдастыру**

##### **A.3.1.4.1 Функционалды диаграммалар**

Rational Rose – UML әдістемесін тарату аспабы. Ұқсас есептерді шешу үшін қолданылатын басқа да визуальды аспаптар. Өңдеу функционалдығын сипаттау – қолдану варианттар диаграммасы, функция орындалу реттері – әрекеттердің тізбектер диаграммасы (талаптар және шектеулер), өңдеу элементтерінің бір-бірімен байланыс және бар болу сипаттамасы – кооперативті диаграммалар (талаптар және шектеулер). Интерфейсті жобалау. Объектілік жобаның өңделуі, ОМ 6 қосымшасына сәйкес, негізінен «прецеденттер диаграммасы» немесе екінші аталуы – «Қолдану варианттар диаграммасы» (КВД) басталады. Прецеденттер диаграммасы барлық қалған жоба диаграммалары үшін арналып жасалады. Ал оның екінші аталуы, негізінен, оның атқаратын рөлін толық көрсетеді деп айтуға болады – пайдаланушы жүйесінің қолдану варианттар диаграммасы (актерлар, әрекет жасаушы объект).

## Б Қосымшасы

(міндетті)

Бағдарлама мәтіні

```
//раволф.php
<?php
defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');
require(__DIR__ . '/../vendor/autoload.php');
require(__DIR__ . '/../vendor/yiisoft/yii2/Yii.php');
require(__DIR__ . '/../common/config/bootstrap.php');
require(__DIR__ . '/../config/bootstrap.php');
$config = yii\helpers\ArrayHelper::merge(
    require(__DIR__ . '/../common/config/main.php'),
    require(__DIR__ . '/../common/config/main-local.php'),
    require(__DIR__ . '/../config/main.php'),
    require(__DIR__ . '/../config/main-local.php')
);
(new yii\web\Application($config))->run();
<?php
use yii\helpers\Html;
use yii\widgets\ListView;
/* @var $this yii\web\View */
/* @var $searchModel frontend\models\UserSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */
$this->title = 'Қолданушылар';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="user-index">
    <h1><?= Html::encode($this->title) ?></h1>
    <?php echo $this->render('_search', ['model' => $searchModel]); ?>
    <?= ListView::widget([
        'dataProvider' => $dataProvider,
        'itemOptions' => ['class' => 'item'],
        'itemView' => '_item',
    ]) ?>
</div>
<?php
use yii\helpers\Html;
use yii\widgets\ListView;
/* @var $this yii\web\View */
```

## Б Қосымшасының жалғасы

```
/* @var $user frontend\models\User */
/* @var $searchModel frontend\models\PostSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */
$this->title = 'Бөлшектер';
$this->params['breadcrumbs'][] = ['label' => 'Қолданушылар', 'url' =>
['users/index']];
$this->params['breadcrumbs'][] = ['label' => $user->username, 'url' => ['users/view',
'id' => $user->id]];
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="post-index">
    <h1><?= Html::encode($this->title) ?></h1>
    <?= ListView::widget([
        'dataProvider' => $dataProvider,
        'itemOptions' => ['class' => 'item'],
        'itemView' => '_item',
    ]) ?>
</div>
<?php
use yii\helpers\Html;
/* @var $this yii\web\View */
$this->title = 'Таптым';
?>
<div class="site-index">
    <div class="jumbotron">
        <h2>!</h2>
        <p class="lead">.</p>
        <p><a
            class="btn
                btn-lg
                btn-success"
href="https://drive.google.com/file/d/0Bz46-
9FqpJZBWE85d3hVdUI4UG9kTWs3VTcyeS1KN2ZPUjI0/view?usp=sharing">Мо
бильді қосымшамен қолданыңыздар</a></p>
    </div>
    <div class="body-content">
        <div class="col-lg-4">
            <h2>Camry</h2>
            <p><?=
                Html::img('http://www.funbook.kz/web/images/madeni.jpg',
['width'=>'350px']);?></p>
            <p>.</p>
            <p><a
                class="btn
                    btn-default"
href="http://www.yiiframework.com/extensions/">Yii Extensions &raquo;</a></p>
        </div>
        <div class="col-lg-4">
```

## Б Қосымшасының жалғасы

```
<h2>Веранда</h2>
<p><?=      Html::img('http://www.funbook.kz/web/images/veranda.jpg',
['width'=>'350px']);?></p>
<p></p>
<p><a
                                class="btn
                                btn-default"
href="http://www.yiiframework.com/extensions/">Yii Extensions &raquo;</a></p>
</div>
<div class="col-lg-4">
  <h2>Плюшкин</h2>
  <p><?=      Html::img('http://www.funbook.kz/web/images/plushkin.jpg',
['width'=>'350px']);?></p>
  <p></p>
  <p><a
                                class="btn
                                btn-default"
href="http://www.yiiframework.com/extensions/">Yii Extensions &raquo;</a></p>
</div>
</div>
</div>
<?php
use yii\helpers\Html;
use yii\widgets\DetailView;
/* @var $this yii\web\View */
/* @var $model common\models\User */
$this->title = 'Қолданушы жайлы';
$this->params['breadcrumbs'][] = ['label' => 'Қолданушылар', 'url' =>
['users/index']];
$this->params['breadcrumbs'][] = ['label' => $model->username, 'url' =>
['users/view', 'id' => $model->id]];
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="user-view">
  <h1><?= Html::encode($this->title) ?></h1>
  <p>
    <?= Html::a('Өңдеу', ['update'], ['class' => 'btn btn-primary']) ?>
    <!--
    <?= Html::a('Жою', ['delete'], [
      'class' => 'btn btn-danger',
      'data' => [
        'confirm' => 'Are you sure you want to delete this item?',
        'method' => 'post',
      ],
    ])?>
  </p>
  <!--!>
```

## Б қосымшасының жалғасы

```
</p>
<?= DetailView::widget([
    'model' => $model,
    'attributes' => [
        'username',
        'email:email',
        'created_at:datetime',
        'description:ntext',
    ],
]) ?>
<?= Html::img($model->image, ['width'=>'500px']);?>
</div>
<?php
use yii\helpers\Html;
use yii\widgets\ListView;
/* @var $this yii\web\View */
/* @var $searchModel frontend\models\PostSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */
$this->title = 'Бқлшекте';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="post-index">
    <h1><?= Html::encode($this->title) ?></h1>
    <?php echo $this->render('_search', ['model' => $searchModel]); ?>
    <?= ListView::widget([
        'dataProvider' => $dataProvider,
        'itemOptions' => ['class' => 'item'],
        'itemView' => '_item',
    ]) ?>
</div>
<?php
/* @var $this \yii\web\View */
/* @var $content string */
use yii\helpers\Html;
use yii\bootstrap\Nav;
use yii\bootstrap\NavBar;
use yii\widgets\Breadcrumbs;
use frontend\assets\AppAsset;
use common\widgets\Alert;
AppAsset::register($this);
?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
```



## Б Қосымшасының жалғасы

```
<html lang="<?= Yii::$app->language ?>">
<head>
    <meta charset="<?= Yii::$app->charset ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <?= Html::csrfMetaTags() ?>
    <title><?= Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>
<body>
<?php $this->beginBody() ?>
<div class="wrap">
    <?php
    NavBar::begin([
        'brandLabel' => 'Auto',
        'brandUrl' => Yii::$app->homeUrl,
        'options' => [
            'class' => 'navbar-inverse navbar-fixed-top',
        ],
    ]);
    $menuItems = [
        ['label' => 'Бастысы', 'url' => ['/site/index']],
        ['label' => 'Бөлшектер', 'url' => ['/posts/index']],
        ['label' => 'Қолданушылар', 'url' => ['/users/index']],
    ];
    if (Yii::$app->user->isGuest) {
        $menuItems[] = ['label' => 'Тіркелу', 'url' => ['/site/signup']];
        $menuItems[] = ['label' => 'Кіру', 'url' => ['/site/login']];
    } else {
        $menuItems[] = ['label' => Yii::$app->user->identity->username, 'url' =>
        ['/users/view', 'id' => Yii::$app->user->id]];
        $menuItems[] = ['label' => 'Қолданушы жайлы', 'url' => ['/profile/index']];
        $menuItems[] = '<li>'
            . Html::beginForm(['/site/logout'], 'post')
            . Html::submitButton(
                'Шығу',
                ['class' => 'btn btn-link logout']
            )
            . Html::endForm()
            . '</li>';
    }
    echo Nav::widget([
        'options' => ['class' => 'navbar-nav navbar-right'],
```

## Б Қосымшасының жалғасы

```
'items' => $menuItems,
]);
NavBar::end();
?>
<div class="container">
    <?= Breadcrumbs::widget([
        'links' => isset($this->params['breadcrumbs']) ? $this->params['breadcrumbs']
: [],
        ]) ?>
    <?= Alert::widget() ?>
    <?= $content ?>
</div>
</div>
<footer class="footer">
    <div class="container">
        <p class="pull-left">&copy; Auto <?= date('Y') ?></p>
        <p class="pull-right"><?= Yii::powered() ?></p>
    </div>
</footer>
<?php $this->endBody() ?>
</body>
</html>
<?php $this->endPage() ?>
<?php
namespace frontend\controllers;
use Yii;
use common\models\User;
use yii\filters\AccessControl;
use yii\filters\VerbFilter;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\web\UploadedFile;
    Yii::$app->session->setFlash('success', 'New password saved.');
```

```
        return $this->goHome();
    }
    return $this->render('resetPassword', [
        'model' => $model,
    ]);
}
}
}
<?php
namespace frontend\controllers;
```

## Б Қосымшасының жалғасы

```
use common\rbac\Rbac;
use common\models\User;
use Yii;
use common\models\Post;
use yii\data\ActiveDataProvider;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\ForbiddenHttpException;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;
class UserPostsController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['create', 'update', 'delete'],
                'rules' => [
                    [
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }
    public function actionIndex($user_id)
    {
        $user = $this->findUserModel($user_id);
        $dataProvider = new ActiveDataProvider([
            'query' => Post::find()->forUser($user->id)->orderBy(['id' => SORT_DESC]),
        ]);
        return $this->render('index', [
            'user' => $user,
            'dataProvider' => $dataProvider,
        ]);
    }
}
```

## Б Қосымшасының жалғасы

```
    );
}
public function actionView($user_id, $id)
{
    return $this->render('view', [
        'user' => $this->findUserModel($user_id),
        'model' => $this->findPostModel($user_id, $id),
    ]);
}
public function actionCreate($user_id)
{
    $user = $this->findUserModel($user_id);
    if ($user->id != Yii::$app->user->id) {
        throw new ForbiddenHttpException('Forbidden.');
```

```
    }
    $model = new Post();
    $model->user_id = $user_id;
    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'user_id' => $user->id, 'id' => $model->id]);
    } else {
        return $this->render('create', [
            'user' => $user,
            'model' => $model,
        ]);
    }
}
public function actionUpdate($user_id, $id)
{
    $user = $this->findUserModel($user_id);
    $model = $this->findPostModel($user_id, $id);
    if (!Yii::$app->user->can(Rbac::MANAGE_POST, ['post' => $model])) {
        throw new ForbiddenHttpException('Forbidden.');
```

```
    }
    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'user_id' => $user->id, 'id' => $model->id]);
    } else {
        return $this->render('update', [
            'user' => $user,
            'model' => $model,
        ]);
    }
}
}
```

```

public function actionDelete($user_id, $id)
{
    $user = $this->findUserModel($user_id);
    $model = $this->findPostModel($user_id, $id);
    if (!Yii::$app->user->can(Rbac::MANAGE_POST, ['post' => $model])) {
        throw new ForbiddenHttpException('Forbidden.');
```

```

    }
    $model->delete();
    return $this->redirect(['index', 'user_id' => $user->id]);
}
/**
 * @param integer $user_id
 * @return User the loaded model
 * @throws NotFoundHttpException
 */
private function findUserModel($user_id)
{
    if (($model = User::findOne($user_id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
```

```

    }
}
/**
 * @param integer $user_id
 * @param integer $id
 * @return Post the loaded model
 * @throws NotFoundHttpException
 */
private function findPostModel($user_id, $id)
{
    if (($model = Post::find()->forUser($user_id)->andWhere(['id' => $id])->one()
    !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
```

```

    }
}
}

//
<?php
namespace frontend\controllers;
use Yii;

```

```
use common\models\User;
use frontend\models\UserSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;

class UsersController extends Controller
{
    public function actionIndex()
```



