

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Шадиева Н.К.

Android платформасында “SATTI ZHOL” қалааралық жолаушылар тасымалы  
мобильді қосымшасын әзірлеу

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

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

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

**БЕКІТЕМІН**

ПИ кафедра меңгерушісі,  
канд.техн. наук, доцент

ассистент-профессор

Р. Юнусов

"14" мамыр 2019ж.

**Дипломдық жобаны орындауға**

**ТАПСЫРМА**

Білім алушыға Шадиева Ноила Камалбековна

Тақырыбы: Android платформасында "SATTI ZHOL" қалааралық жолаушылар тасымалы мобильді қосымшасын әзірлеу

Университет ректоры бұйрығының № 1162-6 "16" қазан 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі "14" мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма, ER-диаграмма түрінде ақпаратты сақтауға арналған деректер қорының сипаттамасы.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

- a) пайдаланушы рөлдерін бөлу жүйесін жобалау
- b) тұжырымдамаға сәйкес android-қосымшаның көп деңгейлі архитектурасын енгізу;
- в) пайдаланушы интерфейсін жобалау және дамыту;
- г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):  
презентацияның 12 слайды ұсынылған



Ұсынылған негізгі әдебиеттер: 10 әдебиеттер тізімінен

Дипломдық жобаны орындау

**КЕСТЕСІ**

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Пәндік аумақты талдау, бағдарламалық кешенді жобалау үшін техникалық спецификацияларды әзірлеу	14.01.2019	орындалды
2. Базалық үлгідегі класс құрылымын енгізу	18.01.2019	орындалды
3. Пайдаланушы интерфейсін дамыту, «Әкімші», «Менеджер», «Қауіпсіздік басқармасының бастығы» функцияларын орындау үшін контроллерлерді енгізу,	01.02.2019	орындалды
4. Иерархиялық қоңыраулардың логикалық тізбегін қолдану арқылы өтінім құрылымын әзірлеу	15.02.2019	орындалды
5. Интерфейсті дизайн иерархиялық сілтемелер тізбегін дамыту: «bredcrump». Өтінімді тексеру	18.03.2019	орындалды
6. Дипломдық жобаға түсіндірме жазба жазу	26.04.2019	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	А.Рамазан тьютор	13.05.19	
Бағдарламалық бөлім	А.М.Байгаринов магистр техн.наук, лектор	13.05.19	

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

Тапсырманы орындауға қабылдап алған студент \_\_\_\_\_ Н.Шадиева.

Күні

« 3 » 10 2018ж.

## КІРІСПЕ

Технология даму заманында адам өз уақытын үнемдеуі күнделікті жетістіктердің біріне айналды. Уақытты тиімді пайдалануда қазіргі қолданыстағы бағдарламалар көп септігін тигізуде. Бағдарлама қолданушылары кез келген уақытта, тиісті ақпараттарды алып, онлайн тапсырыс беріп, қанша адамдар қалайша сайтқа тіркелуі, бір күнде қанша автобустар жолға шығатындығы, бос орындар санын көрсетуін қамтамасыздандыру керек. Айта кететін болсақ әр түрлі елді мекендер арасындағы жолаушылар қатынасы тұрақты дамыды. Көлік байланыстары қалалар арасында да дамы бастады. Клиенттерге қызмет көрсету бойынша жұмыстың заманауи шарттары кәсіпорынның қызмет көрсету жүйесі , деректерді көп пайдаланушы режимде өте үлкен жылдамдықпен өңдеу деңгейге жетті. Барлық құжат айналымы компьютерде жүзеге асырылады, және барлық ақпарат арнайы базаларда сақталады. Өкінішке орай, клиенттерге қызмет көрсетудің ынғасыз жерлері бар, олар қазыргы заманға сай бағдарламалармен емес көп жерде ескі бағдарламалардан қолданып яғни қағаз парағы мен қаламды пайдаланады. Бұл, әрине, өнімділіктің кемшілігі. Сондықтан өнімділігі бар мәселелерді шешетін бағдарламаларды жазу қажеттілігі туындайды. Жақсы дизайн жоба қазіргі уақытта қоғамдық тамақтану орындарына, клубтарға және сауда орталықтары ғана қажет емес. Қалалықтардың ең танымал орындарының бірі – автовокзал ғимараты қаладағы кез келген вокзалдың ғимараты-бұл қаланың өзі сәні. Сондықтан басқа қала тұрғындары тамаша автовокзалды көріп, қаланың өзі де барғысы келеді, тіпті олар мұнда жол жүрсе де.

Дипломдық жобаның нәтижесінде бағдарламаға кірген қолданушы ұсынылған келесі пункттерді орындалуын қамтамасыздандыру керек:

- сайтқа тіркелу;
- тиісті ақпараттарды толтыру;
- жеке мәліметтер қалдыру;
- соңғы жаңалықтардан хабардар болу;
- билеттерге онлайн закат беру .

Дипломдық жобаның мақсаты - Алматы қаласында Автовокзал (автобусбекеті) - жолаушылар мен тасымалдаушыларға қызмет көрсетуге арналған Android платформасында “SATTI ZHOL” қалааралық жолаушыларды тасымалдау мобильді қосымшасын әзірлеу арнайы бөлінген аумақта орналасқан кешенін қамтыйтын көлік инфрақұрылымының объектісі жасау билеттерге онлайн тапсырыс беру , автобекеттегі автобустарды жолушының өз тандау жасап жолаушыға барлық ынғайлы жақтарын қарастыру жеке кабинетке киру ол жердегі міндеттерді атқару жолаушы яғни клиент мобильді бағдарламадағы міндеттерді атқару .Ол үшін дипломдық жұмыстың мақсатын және ынғайлы жақтарын қарастыру.Қызметтің нәтижесі енгізуге және пайдалануға дайын бағдарламалық өнім болуы тиіс.

Мақсатқа жету үшін келесі міндеттер іске асырылуы тиіс:

1. Бағдарламалық жасақтаманы теңшеу;
2. Деректер базасын құру;
3. Интерфейсті жасау;
4. Қосымша жұмыс функцияларын іске асыру;
5. Бағдарламалық өнімді тестілеу.



## 1 ЗЕРТТЕУ ОБЪЕКТИСІНІҢ ЖАЛПЫ СИПАТТАМАСЫ

Сайран автобектін не үшін алғаным туралы айта кететін болсам, Алматы қаласында қаншама адам жол жүреді. Жоқ дегеннің өзінде Алматы қаласынан Шымкент қаласына күніне 5-6 автобус жолға шығады . Олардың ішінде басқа қалаларды басқа мемлекеттерден келген жолаушылар да аз емес ендеше неге алматы қаласындағы автобекет автовокзал ды өзгертін олардың бағдарламаларына керекті мәлеметтерді еңгізмеске деп ойламын. Жоқ, әріне бар өте көп ынғайлы жерлері бар, бірақ та Алматы қаласында неге осы кезге дейін Android платформасында мобильды қосымша бағдарламасын жасап шығармады.Айта кететін жағдай жасап шығаруға клиенттерге тағыда басқа шет ел азаматтарына басқа жағынан ынғайлы жасап қоюға болмайды деп ойлаған оймен сол бағдарламаны өз қолыма алдым. Сайран автобекетінде клиенттерге барлық жақтарынан көп ынғайлы жерлері бар бырақта 1993жылдан бастап келе жақта деректер қоры әлі күнге дейін жұмыс атқарып кере жатыр осы кезге дейін мобильды қосымша жасап шығаратын адам жоқ болды, бырақта WEB парақша бар google – сайран автобекет бар парақшада ешқандай ерекшелік жоқ ол парақша 1993жылы Сайран автобекет жұмыс істеу кезінді ғана жасалған ВЕБ парақша. Ал ендеше Android платформасында Android платформасында “SATTI ZHOL” қалааралық жолаушыларды тасмалдау мобильді қосымшасын әзірлеу мобильді қосымшасын ерекшеліктерін айта кетеин болсақ бұл бағдарламаны алғандан басты мақсатым .Алматы каласындаСАЙРАН автобекетінде қалааралық жолаушылар тасымалдау ВЕБ арқалы мен осыны Android платформасында жасап шығуды қаладым. Яғны адамдар клиенттер уақтың унемдеп автовакзалға барып билет тапсырыс берместен оны Google Play қолданбалары (*Google Play Market (плей маркет)*) немесе iPhone App Store мобильды қосымшалар арқылы өзлеріне сақтап алып заказ беруге ынғайлы жақтарын қарастырдым.

### 1.1 Тақырыптың өзектілігі

Қандай да бір аймақ шегінде жолаушыларды тасымалдау бойынша қызметтермен қалалық автовокзалды қамтамасыз етуге арналған. Бұл үшін ұйымдастыру болуы тиіс. Айта кететін жағдай автобекеттер Алматы қаласында бар, САЙРАН автобекетінде қызмет көрсету жөнеді жұмыс жасау өз штатында жүргізушілер, кассирлер, жүйелік әкімшілердің (егер қажет болса), администратор, бухгалтерлер, менеджерлер программистер, кондукторлар, офис-менеджер және тағыда басқа сондай-ақ. Автовокзалдың негізгі клиенттері-осы өңірде тұратын тұлғалар. Енді былайша айткететін болсақ Алматы қаласында екі автобекет бар – «Сайран» халықаралық автобекеті мен «Барлық» қала жанына жүретін автобекет. "Сайран" халықаралық автовокзалы (бұрын орталық автовокзал) — Алматы қаласындағы автовокзал, Қазақстан Республикасындағы ең ірі автовокзалдардың бірі, қалааралық және қала маңындағы рейстерде жолаушыларға қызмет көрсетуге арналған. САЙРАН автовокзалы жұмысының мысалында билет алу үшін жолаушылар телефон арқылы немесе автовакзалға барып билеттерді сатып алады, ал алған билеттерді автобус жүргізушісі ғана биледі. Тапсырыс журналына белгіленеді және қалаға келгеннен кейін кондуктор түскен ақшамен автовакзал кассасына есеп береді, деп айтуға болады. Өкінішке орай, клиенттерге қызмет көрсету өте нашар, олар "ескі бағдарлама" бойынша қағаз парағы мен қаламды пайдаланады. Автобекет Алматы қаласы САЙРАН халықаралық автовокбеті 1983 жылдан бастап жолаушыларды тасымалдау жұмысын атқаруда. Тәулігіне жоқ дегенін өзінде 28-32 мың жолаушы өткізуге жоспарланған, Бұл аумақта Автобекеттің ғимараты, жолаушыларды автобусқа отырғызу орындары, автобустарды техникалық тексеру және шағын жөндеу жұмыстарына арналған орындар бар. Тәулігіне 100 ге дейін автобус сапарға шығады. Автобекетте 51 маршрут, оның ішінде 4 халықаралық, 27 облысаралық және 20 қала маңындағы маршруттар халыққа қызмет көрсетеді. Автобекет 4халықаралық (Урымшы, Инин, Бишкек, Токмак), 32 Облыс аралық (Абай, Астана, Асықата, Аягөз, Баканас, Балхаш, Жаркент, Жезказған, Жибекөжолы, Жетысай, Қарағанды, Қабанбай, Қарғалы, Кентау, Қызылорда, Ленгер, Мақаншы, Мерке, Мойынқұм, Момышұлы, Мырзакент, Өскемен, Павлодар, Сарканд, Сарыағаш, Тараз, Урджар, Учарал, Шардара, Шаульдер, Шолаққорған, Шымкент) 13 облыс ішіндегі (музей Жамбыл, Кастек, Отар, Матыбұлақ, Қапшағай, ст. Ұзынағаш, Дегерес, Бесмойнақ, Шеген, Гвардейский, Ақтерек, Үлгілі, Үнгіртас) және 3 қала аралық



(Ұзынағаш, Мынбаева, Қарғалы) бағыттарға жұмыс істейді. Сайран авто вокзалында тағыда көп ыңғайлы жерлеріде бар өте көп бырақта ескі жол мен жасалынып жатыр Мысалы : Автовокзал батыс бағыттағы қалааралық және қала маңындағы рейстердің жолаушыларына қызмет көрсетуге арналған. Жоба бойынша автовокзалдың өткізу қабілеті тәулігіне 31 мың жолаушыны құрайды. Ашылғаннан бастап 1990 жылдың аяғына дейін автовокзалда жолаушыларға билет сатудың 5 кассасы болды; сағатына 200 жолаушыға арналған кең күту залдары бар. Сайран автовокзалында деректер қорында солай көп қателіктер бар "Сайран" автовокзалының ғимараты жеке меншікте болған жылдары көптеген жолаушылар айналымына және вокзалды пайдаланатын автобустар иелері мен жалға алушылардан алынатын пайдаға қарамастан, іс жүзінде жөнделмеген және техникалық қызмет көрсетілмеген клиенттер автовокзалға барып билет ке тапсырыс береді кейде адамдардың оған уақты болмау да мүмкін . 2004 жылы автовокзал аумағында оның иесі вокзал әкімшілігінің 2006 жылы шетелдік консультанттардың ұсынысы бойынша сол кездегі алматылық әкім Иманғали Тасмағамбетов белсенді қолдау көрсеткен "Сайран" автовокзалының қызметін тоқтату идеясы туындады. Оның айтуынша, автовокзал оңтүстік астанадан тыс жерде орналасуы керек, бұл қала көшелерін жеңілдетеді және экологияны жақсартады, "Сайран" автовокзалы қаланы ластайды, ал оның орнында сауда орталығы пайда болуы тиіс. 2007 жылдың шілде айына дейін "Сайран" автовокзалы жабылмақ, оның орнына Қаскелең қаласының жанында орналасқан "Үш қоңыр" жаңа вокзалы іске қосылуы тиіс. Сонымен қатар, билік Алматы қаласының жолаушылар Жаңа "Үш қоңыр" вокзалы мен қала сыртындағы вокзалға келе жатқан жолаушылар ретінде Алматы қаласына дейін жетулері тиіс екенін нақтылаған жоқ.

## **1.2 Android платформасы туралы негізгі фактілер**

Ешкімге құпия емес, ХХІ ғасыр – ақпараттық ғасыр технологиялар. ИТ біздің өмірімізге берік кірді: олар бізді барлық жерде қоршайды. Бұл қажеттілігі туды ыңғайлы пайдаланушының және барлық қоршаған инфрақұрылымның өзара іс-қимылы. Пайда болды электрониканы басқару пультін пайдалану қажеттілігі. Apple, содан кейін Google инженерлері шешім

тапты. Олар операциялық телефондарға арналған жүйе электрондық кітаптарды экраннан оқу мүмкіндігінен бастап қолданбалар ұялы телефондар және тұрмыстық техниканы "ақылды" басқарумен аяқталған Google компаниясы бұдан әрі өзінің амбициялық жоспарларында болып, оны құрдыашық Android архитектурасы. Android (Android) - портативті (желілік) коммуникаторларға, планшеттік компьютерлерге арналған Операциялық жүйе, электрондық кітапшалар, цифрлық ойнатқыштар, қол сағаттары, нетбуктар және Linux ядросына негізделген смартбуков. Бастапқыда компания әзірлеген Android Inc., содан кейін Google сатып алды.

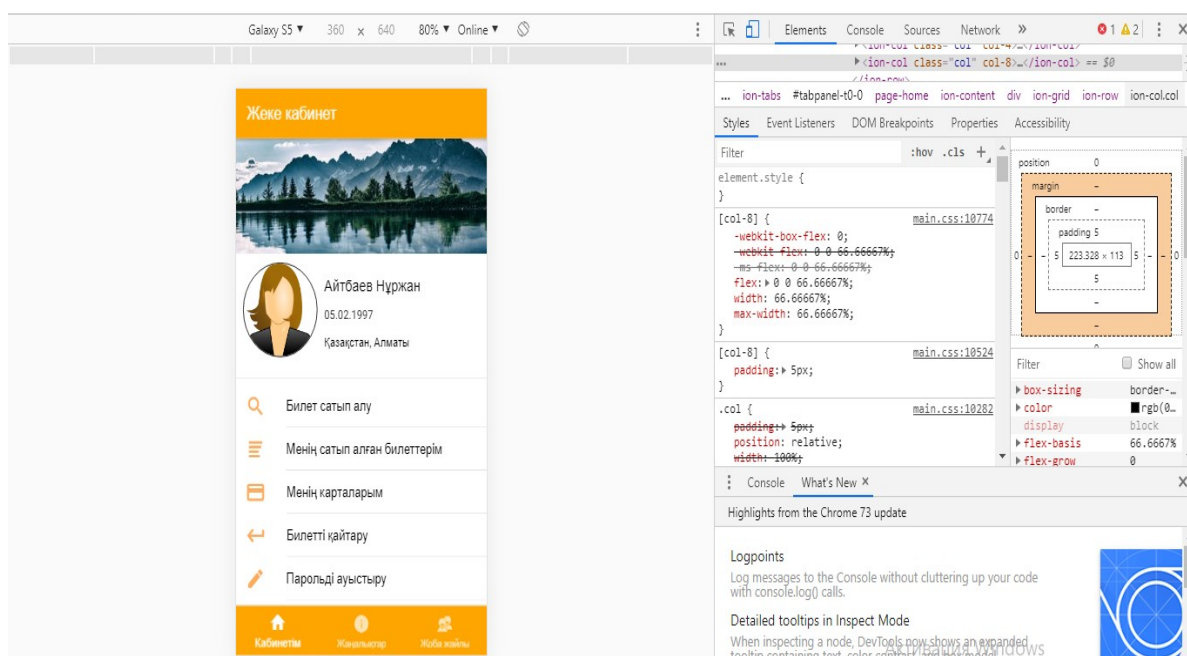
2012 жылдың үшінші тоқсанында сатылған смартфондардың 75 % - дан астамы Android операциялық жүйесімен жабдықталған. Енді әрбір әзірлеуші электрондық құрылғы Android-ті өз астына қайта жасауға мүмкіндігі бар құрылғы, осылайша, өз құрал-жабдықтары мен осы OS үшін үшінші тарап қолданбалары. Бұл өте тиімді болды. Android шығу өз бетінше біреуден жоғалта отырып, операциялық жүйені жазған немесе сатып алған.

Осылайша, барлық бағдарламашылар жасаған пайдалы бағдарламалар Android ОЖ шыққаннан кейін өндірушілер алдында жиі сұрақ туындайды, Android-тың қандай нұсқасын қолдау керек. Android пайда болуы мен қарым-қатынаста маңызды рөл атқарды адамдармен. Оны телефондарға орнату оңай мүмкіндік берді, функционалды кеңейте отырып, мобильді құрылғылардың жаңа модельдерін әзірлеу және операциялық жүйенің өзі қолдайтын болды.

Android платформасы туралы негізгі фактілер мен жасаған бағдарламада дәл осылайша кезлген сайт қа бағдарламаны салып сол жерде ыстеуге болады дай жасалған Android бастапқы коды ашық қол жетімді, сондықтан кез келген адам оны көшіріп, өз нұсқасын өзгертіп, жариялай алады – мүлдем тегін. Бұл Android жабық операциялық жүйелерден ерекшеленеді. Оларға кіру қатаң шектелген, ал әзірлеушілер оларды өзгертуге тыйым салады.. Android барлық нұсқалары ашық қол жетімді, сондықтан құрылғылар мен әзірлеушілер тегін жүктеу және оларды пайдалана алады. Android платформасында Android платформасында “SATTI ZHOL” қалааралық жолаушыларды тасмалдау мобильді қосымшасын әзірлеу арнайы Android платформасына арналған қосымшаларды әзірлеу негізінен жүргізілед. Java тілінде. Java тілінде бағдарламалар жасау үшін арнайы бағдарламалық қамтамасыз ету. Бұл БҚ ең соңғы нұсқаларын жүктеуге болады.

Oracle Corporation ресми сайтынан. Бұл бағдарламалық кешен JRE (Java Runtime Environment) және JDK (Java Development Kit). Бірінші құрал-бұл орта орындау-виртуалды машинаны ең аз іске асыру, онда Java бағдарламалық коды іске қосылады және орындалады. Екінші құрал-бұл өз кезегінде, толық құралдар жиынтығы, қосымшалар әзірлеушісі жиынтығы Java тілінде. Шын мәнінде, JRE сондай-ақ JDK құрамына кіреді Java кластарының әртүрлі стандартты кітапханалары, javac компиляторы, құжаттама, код үлгілері және түрлі қызметтік утилиттер. Бұл барлық жиынтық еркін таралады және OS түрлі нұсқалары бар, сондықтан кез келген оны жүктеуге және пайдалануға болады. JDK интеграцияланған емес әзірлеу ортасы, әзірлеуші белгілейді

Бір бөлек. Java әзірлеу үшін көптеген IDE бар, мысалы, NetBeans, IntelliJ IDEA, Borland JBuilder және басқалар. Осылайша, бұрын Android ОЖ базасында қосымшаны әзірлеуге кірісу қажет құрал-саймандар жасау



1.2.1-сурет - Android платформасында “SATTI ZHOL”

## 2 ЖОБАЛАУ БӨЛІМІ

Автовокзал – жолаушылар тәулігіне бес жүзден астам адамды құрайтын автобустардың жолаушылары мен жүргізушілеріне қызмет көрсетуге арналған құрылыстар кешені және құрамында: билет кассалары, сыйымдылығы жетпіс бестен жоғары жолаушыларға арналған күту залы бар күрделі ғимаратты, жүргізушілерді медициналық тексерістерден өткізуге және алғашқы медициналық-санитариялық көмек көрсетуге арналған медициналық пункттерді, сақтау камераларын, жолаушыларды түсіру мен отырғызуға арналған перронды, автобустарға арналған тұрақтарды, автобустарды тексеріп қарауға арналған бекеттерді, диспетчерлік пункттерді және ана мен балаға арналған бөлмені қамтитын талдау жасау. "АВТОВОКЗАЛ" жүйесі жеке автовокзал, бір қаланың автовокзалдар тобы немесе бүкіл аймақтың қалааралық автобус маршруттарының желісі деңгейінде қалааралық және қала маңындағы автобус тасымалдарына билеттерді сатуды автоматтандыруға арналған. Жүйе келесі мүмкіндіктерді қамтамасыз етеді: тікелей және кері бағытта, тек бір бағытта, ал қажет болған жағдайда маршруттағы аралық станциялардан билет сату, тікелей сату және алдын ала тапсырыс бойынша сату, рейспен жұмыстардың барлық циклын диспетчерлік сүйемелдеу, рейстің ашылуынан сатылуға тиеу ведомосін жүргізушіге бергенге дейін, кассирлердің түсімі туралы бухгалтерияға және автовокзалдардың автоматтандырылған желісіне қызмет көрсететін кәсіпорындармен есеп айырысу үшін деректерді дайындау. "Автовокзал+" автоматтандырылған жүйесі билеттерді есепке алу және сату процестерін автоматтандыруға арналған. Android платформасында "SATTI ZHOL" қалааралық жолаушыларды тасымалдау мобильді қосымшасын әзірлеу.

Автовокзал жұмыс істеу саласы Жалпы пайдаланымдағы автокөлік кәсіпорындары автовокзалдан орындайтын облысшілік, облысаралық және республикааралық қалааралық автобус тасымалдары болып табылады.

"Автовокзал" автобус билеттерін ағымдағы және алдын ала сатудың, қалааралық маршруттардағы автобустардың қозғалысын диспетчерлік басқарудың, өткізуді есепке алудың, есептілікті қалыптастырудың технологиялық процестерін автоматтандырады.

Android платформасында "SATTI ZHOL" қалааралық жолаушыларды тасымалдау мобильді қосымшасын әзірлеу құру нәтижесінде мынадай функциялар автоматтандырылады:

- қалааралық автобус рейстеріне билеттерді алдын ала және ағымдағы сату;
- қалааралық қатынастағы автобустарда орындарды брондау операцияларын жүзеге асыру;
- қабылдау бойынша тапсырыстарды Android платформасында “SATTI ZHOL” қалааралық жолаушылар тасымалдау мобильді қосымшасын бойынша;
- қалааралық автобус маршруттарына жол жүру және багаж билеттерін сату бойынша кассирлердің жұмысы туралы есеп жасау және есеп беру;
- қалааралық автобустардың қозғалыс кестесін жасау;
- автобус рейстерінің келуін және жөнелтілуін бақылау, диспетчерлік есептілікті жасау;
- автовокзалдың автобус қатынасында жолаушылар мен багажды тасымалдаудан, комиссиялық алымдардан кірістерін есепке алу, тиісті есептілікті жасау;
- автокөлік кәсіпорындары (АТП) бөлінісінде жолаушыларды тасымалдаудан түскен кірістерді есепке алу.

Менің ойымша, АВТОВОКЗАЛ бұл өз кезегінде жолаушылар ағыны үлкен емес шағын қалалар үшін кемшілік болып табылады. Бұл жүйені, аппараттық бөлікті сатып алуға үлкен шығындар қажет және персоналды оқытуға көп уақыт қажет. Бірақ та бағдарлама жасау шығаруға болады ол үшін көп нәрсені қарастыру керек. Жасағанның өзінде дайын бағдарлама жасап шығарылады Android платформасында болады, мобильді қосымшасын бойынша. Оған ыстеу қалай жасау қымдер керек барын бағдарламаға қосылады.

Айта кететін болсақ:

Орындаушының 3 міндеттері

Жолаушы-сапарға ақы төлеу билетті қалаи сатып алуы тапырыс беруы;

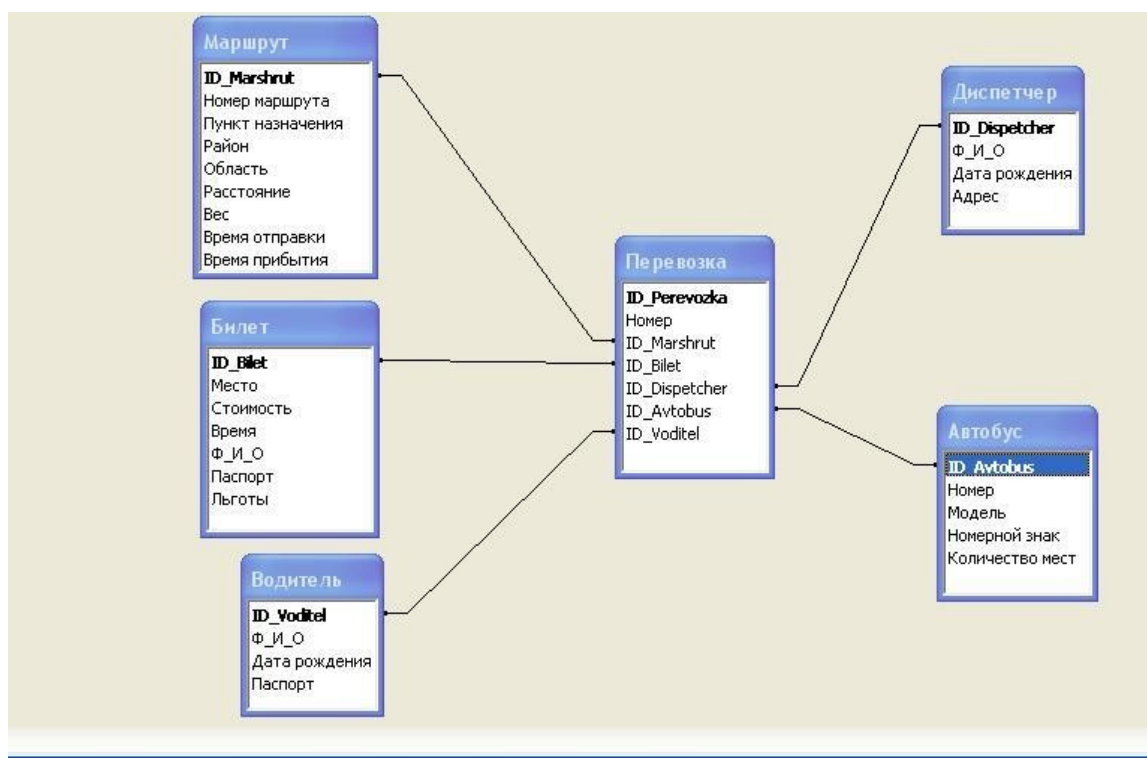
Кассир-клиентке қызмет көрсету;

Администратор -жүйе мен деректер базасына қызмет көрсету(резервтік көшіру, қалпына келтіру, жаңа пайдаланушыларды қосу.

## **2.1 Деректер қорын жобалау**

Android әзірлеушісі жұмыс істей отырып, мобильді қосымшаларда деректер базасын жобалаудың екі түрлі тәсілдеріне тап болдым. Мүмкін бұл жерде біреу анық көрінеді, ал мүмкін біреу жаңа идея береді немесе қателіктерден сақтайды. Жалпы, ұзаққа созбаи іске көшеміз. Android платформасында “SATTI ZHOL” қалааралық жолаушыларды тасмалдау мобильді қосымшасын әзірлеу . Деректер қорын жобалау мақсатында бірінші біз не істеу керек қандай жолмен жасап алуымыз керек екендігін көріп отейік. Бес кесте туралы деректерді қамтиды:

- тасымалдау маршруттарында;
- сатылған билеттерде;
- деректер базасына ақпарат енгізетін диспетчер;
- тасымалдау жүзеге асырылатын көлік құралы
- тасымалдауды орындайтын жүргізуші.

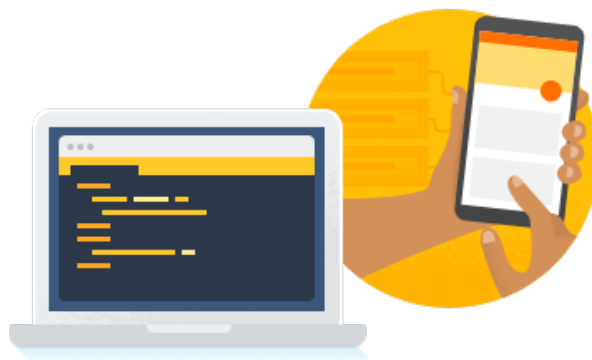


1.2.1-сурет - сурет - Деректер қорының кестелері арасындағы өзара әрекеттесу сызбасы

## 2.2 Деректер қорын құру

Деректер қорын құруда файлының көлемі айтарлықтай болады. Стандартты тәсілді қолданғанда Деректер қорыда файлының көлемі аз. Бұл деректерді сақтау кезінде, қалпына келтіру салдарынан артық болмауымен байланысты. Firebase-дын realtime database деректер қорын құру бағдарламаларынан пайдалана отырып деректер қорын құрдым. Теорияда нормалану дәрежесі жоғары болған сайын, артықтығы аз, алайда, осы деректерді оқу кезінде Деректер қорына жүктеме артады. Көптеген ресурстар кестелердің джойндарына жұмсалады. Key-value тәсілін қолданғанда деректердің артық дәрежесі жоғары, себебі қалыпты деңгейі айтарлықтай аз, бұл Деректер қорын файлының көлемін арттыруға әкеледі. Әдетте жобаның дамуымен Деректерды құру схемасы бір рет өзгертілмейді, жаңа өрістер қосылады, бұрын қолданылған нысандар бірнеше жаңасына ұсақталады немесе керісінше оларды денормалау және бірнеше кестелерді бір кестеге біріктіру жүргізіледі. Егер схеманы жаңарту кезінде біз деректер қорын да жинақталған деректерді алатын алатын болсақ, онда барлығы оңай: біз схеманы жаңартқан сайын жаңа деректер қорын файлын жасаймыз, ал ескісін жойамыз. Бірақ деректер сақталған және жаңа пішімге түрлендірілген болса не істеу керек? Бұл нұсқада стандартты тәсіл артықшылығы бар. Әдетте жобаның дамуымен деректер қорын схемасы бір рет өзгертілмейді, бырақта өзімі үшін өрістер қосыға болады, бұрын қолданылған нысандар бірнеше жаңасына ұсақталады немесе керісінше оларды денормалау және бірнеше кестелерді бір кестеге біріктіру жүргізіледі. Егер схеманы жаңарту кезінде біз деректер қорын жинақталған деректерді өараитын болсақ, онда барлығы оңай: біз схеманы жаңартқан сайын жаңа деректер қорын файлын жасаймыз, ал ескісін жойамыз. Бірақ деректер сақталған және жаңа пішімге түрлендірілген болса не істеу керек? Бұл нұсқада стандартты тәсіл артықшылығы бар. Деректер қорын сұлбасын жаңарту қарапайым міндет емес. Барлық деректерді сақтай отырып, схеманы қайта құру қажет, сондай-ақ жаңа өрістерді инициализациялау және қайта сериалдау арқылы деректерді өзгертіп, деректерді жаңартып отыру қажет. Операцияның логикалық күрделілігі да, жаңартуға қажетті уақыт шығындары да артады. Мен де деректер қорын құру кезінде Firebase-дын realtime database пайдалана отырып жасадым. Айта кететін болсам Firebase әзірлеуші пакеті интуитивті түсінікті API біріктіреді, сізді жеке пакеттерді басқару қажеттілігінен арылту. Сізге қажет нәрсені таңдап, интеграцияланған шешімнің артықшылықтарын

пайдаланыңыз. Google инфрақұрылымын пайдаланатын Платформа әрбір даму және өсу кезеңі үшін қажетті мүмкіндіктер береді.



### 2.2.1-сурет - Firebase

Firestore Realtime деректер қоры - бұл Бұлтты деректер қоры. Деректер JSON форматында сақталады және әрбір қосылған клиентпен нақты уақыт режимінде синхрондалады. IOS, Android және JavaScript үшін біздің SDK көмегімен кроссплатфорлы қосымшалар жасаған кезде, сіздің барлық клиенттеріңіз нақты уақытта деректер қорының бір данасын бірлесіп пайдаланады және автоматты түрде ең жаңа деректер жаңартуларын алады. Firestore танысу уақыты Firestore бетіне өтіп, Firestore-да тіркелгіні тіркеңіз немесе бұрыннан бар бетке кіріңіз. Сондай-ақ, біз GOOGLE тіркелгісімен тіркеле аламыз. Тіркеуден кейін, біз 5 минуттық оқудан өте аламыз, бірақ ол JavaScript жазылған қосымшаларға арналған.

#### Firestore Басты артықшылықтары

Жұмыс істеу жылдамдығы-Firebase әзірлеуші пакетінде интуитивті түсінікті API жиналған, олар сапалы қосымшаларды әзірлеуді жеңілдетеді және жеделдетеді

Дайын инфрақұрылым. Күрделі инфрақұрылым жасау немесе бірнеше басқару панелімен жұмыс жасамасада болады

Статистика-Firebase негізінде ұялы құрылғылар үшін арнайы әзірленген. Firestore үшін Google Analytics пайдаланушыға әрекеттері туралы



алуға және қосымша мүмкіндіктер арқылы шаралар қабылдауға мүмкіндік береді.

- Кроссплатформенность- Firebase Android, iOS, JavaScript және C++ үшін әзірлеушінің пакеттері арқасында кез келген платформаларда жұмыс істейді. Сіз сондай-ақ серверлік кітапханалар немесе REST API арқылы Firebase-ге жүгіне аласыз.

Масштабтау тегін қолдау-Электрондық пошта арқылы тегін қолдау. Сонымен қатар, Firebase командасы мен Google әзірлеу мамандары Stack Overflow және GitHub ресурстарындағы сұрақтарыңызға жауап береді.

Нақты уақыт дерекқоры — БДРВ) - деректерді өңдеу нақты уақыт принципі бойынша жүргізілетін деректер базасы.

ТП баж өнеркәсіптік автоматтандыру жүйелерінде қолданылады. Деректерді синхрондауды, репликациялауды қамтамасыз етуі және уақыттың нақты ауқымында істен шығуға төзімділікті қамтамасыз ету үшін резервтеуді қамтамасыз етуі тиісті программалардың пайдалана отырып жасадым.

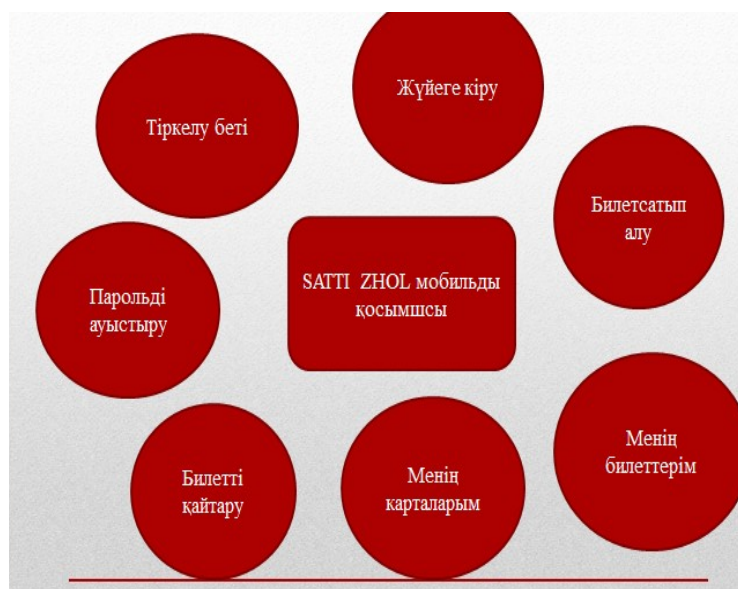
## 2.3 Функциональдық диаграммалар

Мен диаграммалармен жұмыс жасаған кезде былайша айтатын болсақ диаграмма бұл жасап жап жатқан мобильды қосымшасын әзірлеу бағдарламасының жасаған кезде сол диаграммалар мен жұмыс жасап яғни тусіндіріп кетсем макет сияқты жасалады бағдарламаны жасау кезіндегі басты мақсат деуге де болады. Функциональды диаграммалар кезінде мен өзім бағдарламамен емес керекті бағдарламаларды алып солар арқылы диаграмманы жасап шықтым. Диаграмманы жасау кезінде макет сияқты жұмысты мәніне жетіп жасауда болады, айтарлықтай бағдарламалардан пайдалана отырып жұмыс жасау кезінде пакет сияқты жасауда болады маған сол оңай болды. Басында айтып кеткендей деректер қорыдан пайдалана отырып диаграмманы шығарып алдым, тусынтырып айтып кететін болсам диаграмма кезінде былайша мен көптеген диаграммаларды салыстыра отырып өзімді жасаған диаграмма ерекше болуын қаладым, бырықта айта кететіндей ешқандай ерекшелік жоқ Android платформасында “SATTI

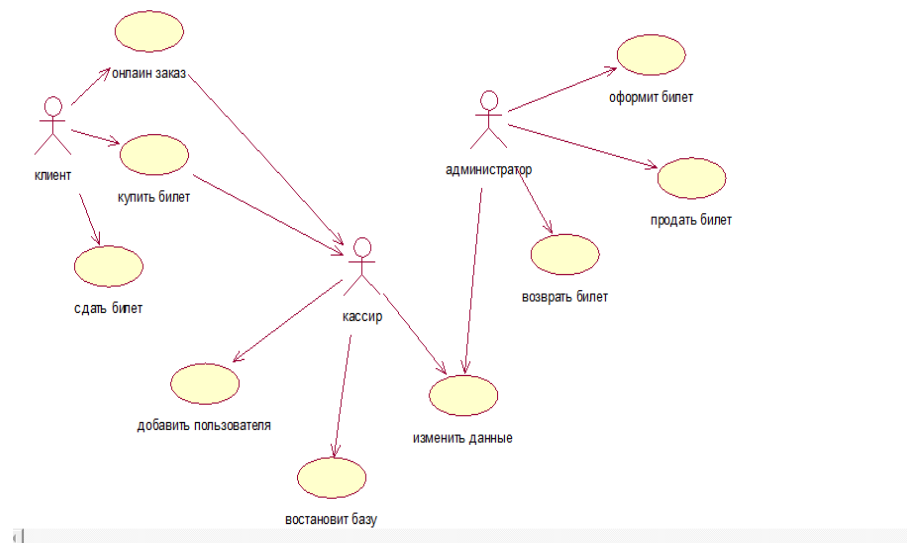
ZHOL” қалааралық жолаушыларды тасмалдау мобильді қосымшасын әзірлеу диаграммасы мен үшін ерекше жағдай болды ,себебі мен өзім бұл бағдарламаны таңдаған себебім Алматы қаласында автобекет САЙРАН қалааралық жолаушылар тасмалдаушы бекеті диаграммасы мен салыстыра отырып жасауды жөн көрдім. Ал ендеше даграммамен танысатын болсақ диаграмманы жасау жолдары мен танысайқ.

Бірінші жай ғана қалааралық жолаушылар “SATTI ZHOL” мобильді қосымшасында макент сияқты шығардым. Функционалды диаграммалар бірінші кезекте әзірленетін бағдарламалық қамтамасыз ету функцияларының өзара байланысын көрсететін диаграммалар қалайша қандай жағдайда бір бірімен байланысуын көрсетеді.

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 варинаттар диаграммасы» (ҚВД) 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).



### 2.3.1-сурет - IDEF0-диаграммасы

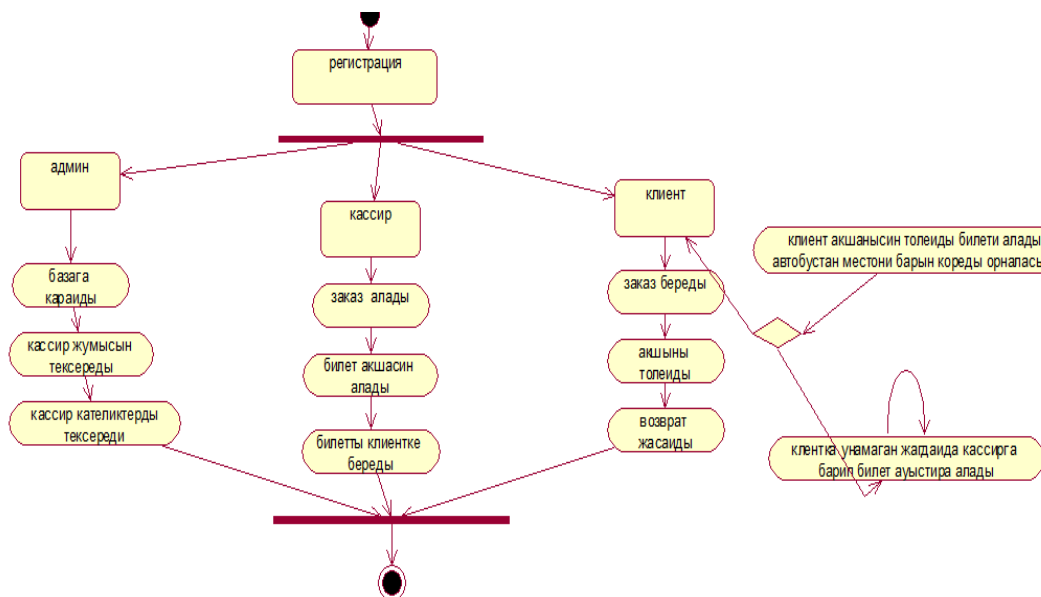


### 2.3.2-сурет - Прецеденттер диаграммасы

Прецеденттердиаграммасы UML диаграммасының әр түрлі жүйелерді талдау үшін жиі пайдаланылады. Олар жүйеде әртүрлі рөлдерді бейнелеуге және осы рөлдермен жүйемен өзара әрекеттесуге мүмкіндік береді.

Диаграммалар жүйесін пайдалану талаптарын жинау үшін қолданылады. Сіздің талаптарыңызға байланысты бұл деректерді әр түрлі жолдармен пайдалануға болады. Оларды пайдаланудың бірнеше жолы бар:

- рөлдердің функциялары мен өзара әрекеттесуін анықтау;
- жоғары деңгейдегі жүйеде таныстыру үшін;
- менеджерлерге немесе мүдделі тұлғаларға ұсынғанда әсіресе пайдалы жағы.



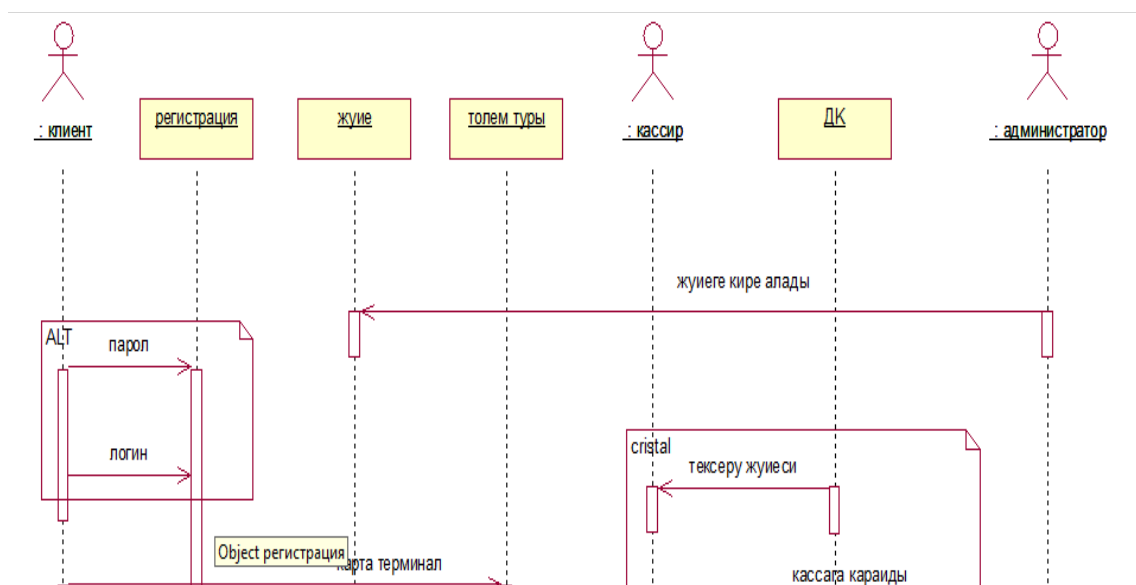
**2.3.3-сурет - Күй диаграммасы**

Күй диаграммасы визуалды түрде блок-схемаға немесе деректер ағынының диаграммасына ұқсас жүйедегі жұмыс үрдісін немесе басқару ағынын білдіреді. Операциялық диаграммалар жиі бизнес-процестерді модельдеуде қолданылады.

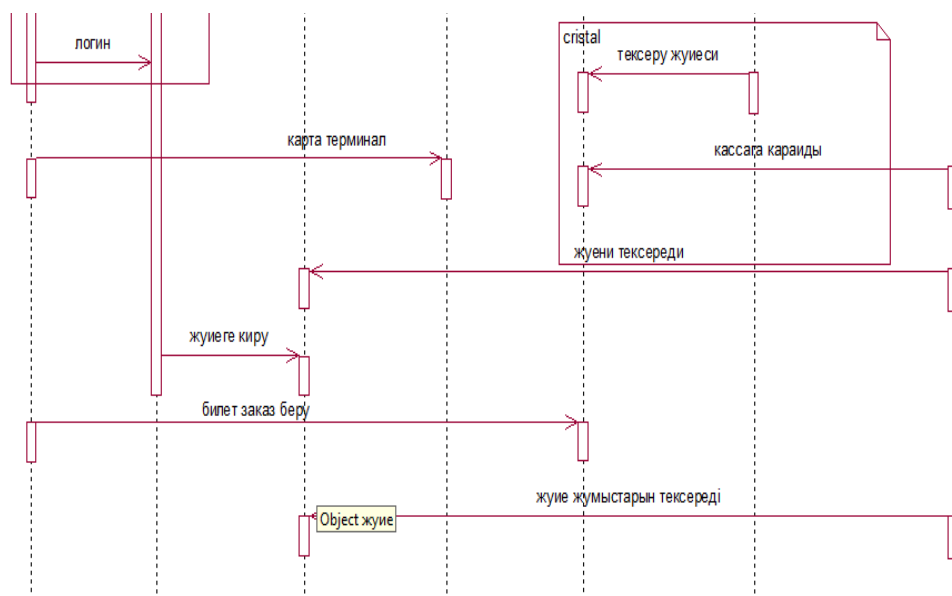
### **Тізбек диаграммасы**

Тізбек диаграммасы – нысандардың бір-бірімен және қандай тәртіппен жұмыс жасайтынын көрсететін өзара әрекеттесу сызбасы. Бұл хабарлама тізбегінің диаграммасының құрылысы. Тізбек диаграммасы уақыт тәртібінде реттелген нысандардың өзара әрекетін көрсетеді. Ол сценарийге қатысатын нысандарды және класстарды көрсетеді және сценарийдің функционалдығын орындау үшін қажетті нысандар арасында алмасатын хабарлар тізбегін

көрсетеді. Схема диаграммалары, әдетте, әзірленген жүйенің логикалық көрінісінде пайдалануды жүзеге асырумен байланысты. Кездейсоқ диаграммалар кейде оқиға диаграммалары немесе оқиға сценарийлері деп аталады.



2.3.4-сурет Тізбек диаграммасы

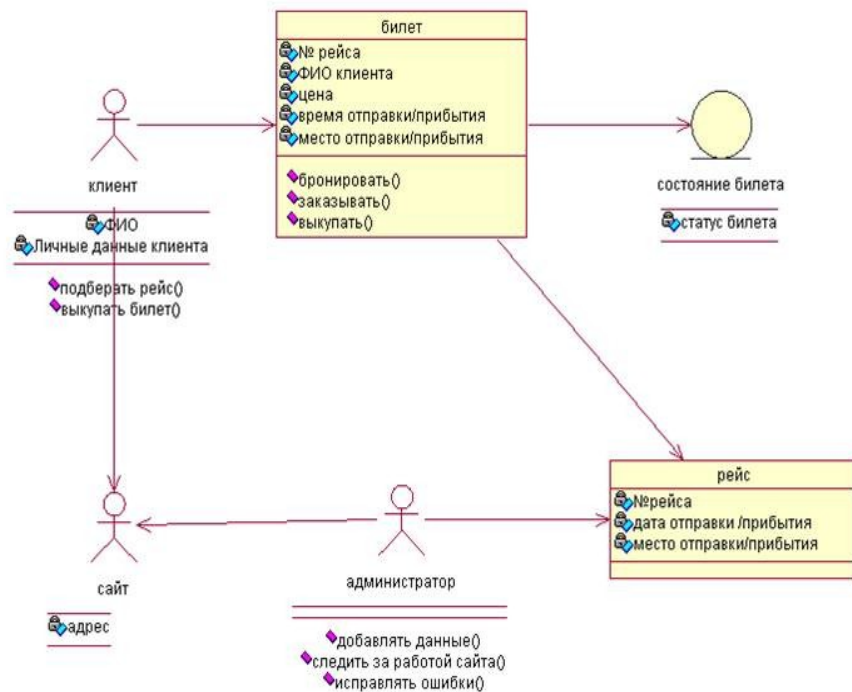


### 2.3.5-сурет Компонент диаграммасы

Компонент диаграммасы – қатынастарды көрсетеді, модульдер арасындағы байланыстар (логикалық немесе физикалық), оның ішінде имитациялық жүйеден тұрады. Компоненттік диаграммадағы субъектілердің негізгі мәні болып табылады, сондай-ақ интерфейстерді көрсетеді компоненттер арасындағы байланыстарды анықтайды. Компонент диаграммада, келесі қатынастар орындалады;

- компоненттер мен интерфейстер арасындағы байланысты іске қосу (интерфейсті жүзеге асырушы);

-компоненттер мен интерфейстер арасындағы тәуелділіктер (интерфейсті қолданады).



### 2.3.6-сурет - Компоненттер

## **Логикалық құрылымы**

Логикалық құрылымы кезінде кез келген бағдарламада болу керек деп ойламын жұмыстын қысқаша тусындыру керек деп ойлап логикалықта қосып кеттім .



**2.3.7-сурет - Логикалық құрылымы**



### 3 БАҒДАРЛАМАЛАУ

Бұл бағдарлама мен үшін қыйында оңайда болған жоқ бырақта ең қызықты сәт дәл осы жер болды. Бағдарламалау болды .Өте көп бағдарламалау тілі ден пайдаландым мен осы технологияларды колданып жасаймын:

- ionicframework
- cordova
- angular
- android studio

IonicFramework -ГИБРИДТІ МОБИЛЬДІ ҚОСЫМШАЛАРДЫ ӘЗІРЛЕУ Ionic-Бұл HTML, CSS және JavaScript Мобильді қосымшаларды әзірлеу платформасы. Иондық қосымшалар жеке қосымшалар ретінде орындалады және жеке "сыртқы түрі"болады.

IonicFramework-бұл веб-технологияларды пайдаланып гибриді Мобильді қосымшаларды әзірлеу үшін ашық бастапқы коды бар SDK//. <https://www.youtube.com/playlist?list=PLwc5ZF4d9kvtnXPHNY0pb6zAMzjPmBStJ>

Cordova- Apache Cordova-ашық бастапқы коды бар мобильді қосымшаларды әзірлеу платформасы. Ол HTML5, CSS3 және JavaScript сияқты стандартты веб-технологияларды қолдануға мүмкіндік береді кросс платформалық дамыту үшін, әрбір ұялы платформалар үшін ана тілін дамыту болдырмау. Қосымшалар әр платформаға бағытталған орам ішінде орындалады және құрылғының датчиктеріне, деректер мен желі жағдайына қол жеткізу үшін Стандартты API-ге сүйенеді.

Егер сіз Apache Cordova:

- мобильді әзірлеуші және қосымшаны бір платформаларды әзірлеудің әрбір тілі мен құралдар жиынтығы үшін оны қайта іске асырудың қажеті жоқ көп платформаларға ұлғайтқыңыз келеді.

- веб-әзірлеуші және түрлі қолданбалар дүкендерінде тарату үшін оралған веб-бағдарламаны ашу керек.

- WebView (арнайы браузер терезе) бар жеке қолданба компоненттерін араластыруға мүдделі мобильді әзірлеуші құрылғының API деңгейіне қол жеткізуге болады, немесе сіз туған және WebView компоненттерінің арасындағы плагин интерфейсін жасау керек болса.

Angular -ашық бастапқы коды бар фреймворк. Бір беттік қосымшаларды әзірлеуге арналған. Оның мақсаты-MVC-үлгі негізінде браузерлік қосымшаларды кеңейту, сондай-ақ тестілеу мен әзірлемелерді жеңілдету.

Фреймворк жұмыс істейді HTML қамтитын қосымша пайдаланушылық атрибуттар, олар сипатталады директиваларын байланыстыратын, енгізу немесе шығару саласындағы беттің моделімен білдіретін жай айнымалылар JavaScript. Бұл айнымалылардың мәндері қолмен беріледі немесе статикалық немесе динамикалық JSON деректерінен алынады.

Басты мақсат бойынша бағдарлама бастаған кездің өзінде мен үшін ерекше орын алып келе жатқан бағдарлама бұл Android studio .

Android Studio-бұл 2013 жылдың 16 мамырында Google I/O конференциясында анонсталған Android платформасымен жұмыс істеу үшін біріктірілген даму ортасы (IDE).

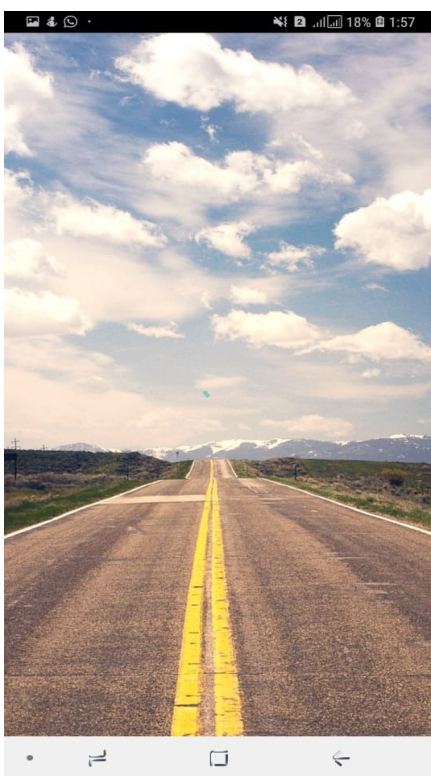
IDE 2013 мамыр айында жарияланған 0.1 нұсқасынан бастап, содан кейін 2014 жылдың маусым айында шығарылған 0.8 нұсқасынан бастап бета-тестілеу сатысына өтті. 1.0 бірінші тұрақты нұсқасы 2014 жылдың желтоқсанында шығарылды, сонда Eclipse үшін Android Development Tools (ADT) плагинін қолдау тоқтатылды.

JetBrains компаниясынан IntelliJ IDEA бағдарламалық жасақтамасына негізделген Android Studio - Android қосымшаларын әзірлеудің ресми құралы. Бұл жұмыс ортасы Windows, OS X және Linux үшін қол жетімді. 17 мамыр 2017, Google I / O жыл сайынғы конференциясында, Google Java және C++ қосу Android платформасына арналған ресми бағдарламалау тілі ретінде Android Studio-да қолданылатын Kotlin тілін қолдауды жариялады.

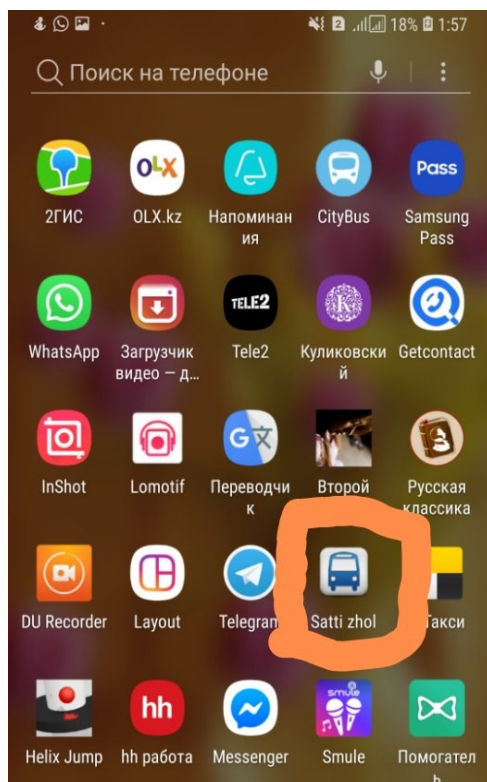
#### **4 ҚОЛДАНБАЛЫ БӨЛІМ**

## 4.1 Жеке кабинет

Осы барлық жұмысты бітіріп болғаннан кейін біз қолданбалы бөлімге жеттік. Жасаған бағдарламада мен барлық жағдайда қалай жұмыс жасаған болсам солай бағдарламаны техникалық және әлеуметтік мәселелерді практикалық шешуге бағытталған ғылыми зерттеулер жасадым.



4.1-сурет Жеке кабинетте кіру



#### 4.2-сурет - ұялы телефондағы қосымша

Ал ендеше айта кететін болсақ Android платформасында “SATTI ZHOL” қалааралық жолаушылар тасмалдау мобильді қосымшасында бағдарламаға кірген кезде қалай не истеу керек екенін айта кететін болсам

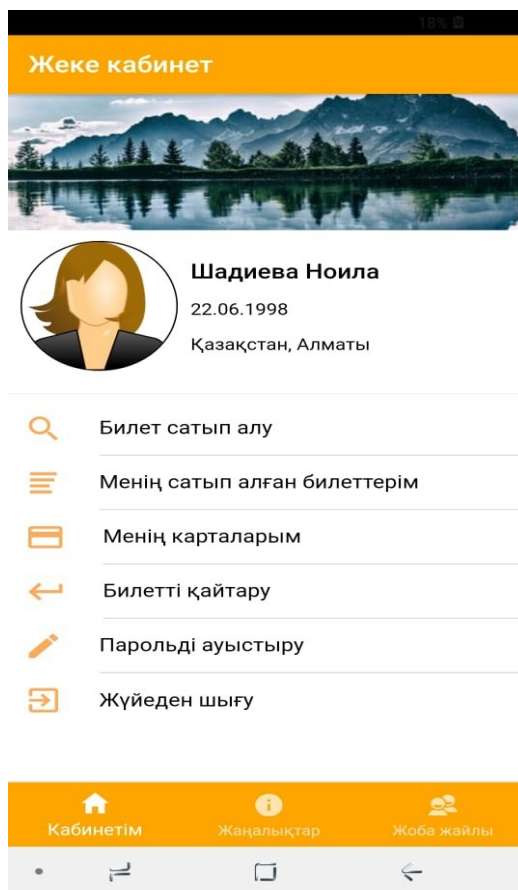
4.1 Жеке кабинет

4.2 Билет сатып алу

4.3 Менің карталарым

4.4 Жаңалықтар

Жеке кабинет айта кететін болсақ мен “SATTI ZHOL” бағдарламасында адамдар тапсырыс бері үшін бірінші болып ЖЕКЕ КАБИНЕТ кіруі керек сол жерді өздерінің парол логен жасып тіркеліп “SATTI ZHOL” бағдарламасына кіре алады

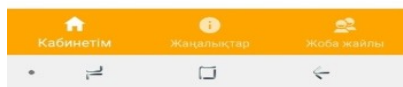
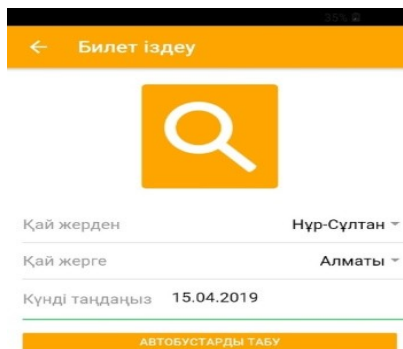


**4.3-сурет** - Басты бет

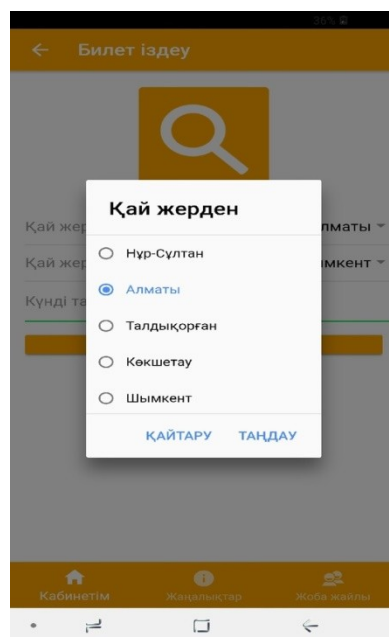
## 4.2 Билет сатып алу

Жеке кабинетке кірген адам билеттірді “SATTI ZHOL” бағдарламасына кіреді содан кейін барып билеттер туралы ақпараттар ала алады , айтарлықтай ондай күшті бағдарлама болмасада адамдарды уақтын унемдей алатындай бағдарлама билеттерге тапсырыс бері үшін адамдар өздеріне керекті қалаларды тандай алады орындарды сумма тандай алады . Одан кейін барып билеттерге тапсырыс бергенден кейін барлық жарттарды орындап билетті берілген код автобекетке көрсетіп жолға шығады.

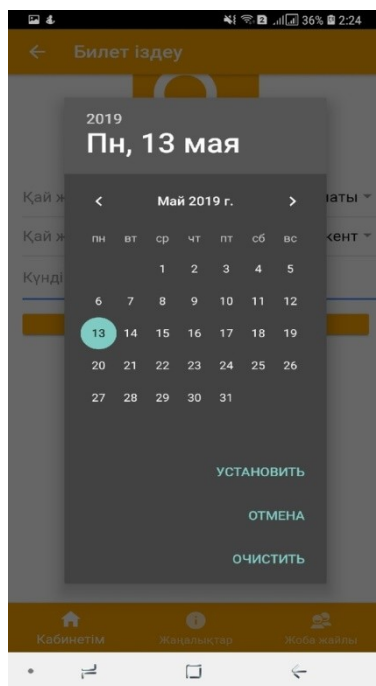
Бұл суретте Билетті сатып алу үшін қай қалаға кететініміз туралы мәлімет яғни толдырамыз (Нұр-Сұлтан-Алматы)



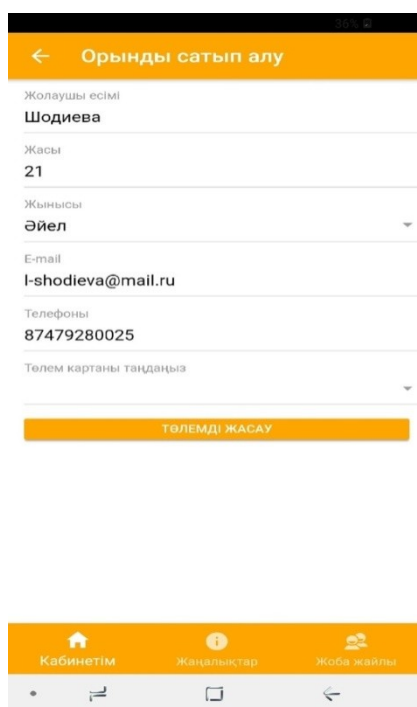
4.4-сурет Билеттері іздеу



4.5.-сурет Қалаларды таңдау



4.5-сурет - Календар бөйінша қарастыру



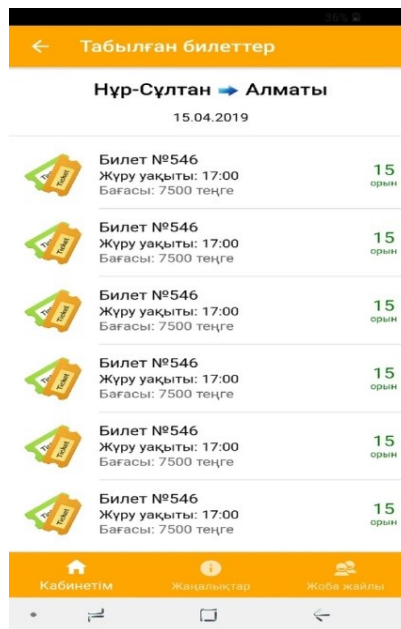


#### 4.7-сурет - Календар бөйінша қарастыру

Уақтың қай күні кететіндігін таңдағандан кейін ОРЫНДЫ САТЫП АЛУ туралы ақпаратқа көшеміз ол жерде барлық бар нәрсені толдырамыз . Содан кейін барып ғана отыратын орынды таңдаймыз С1ден С12 және В1 ден В12.

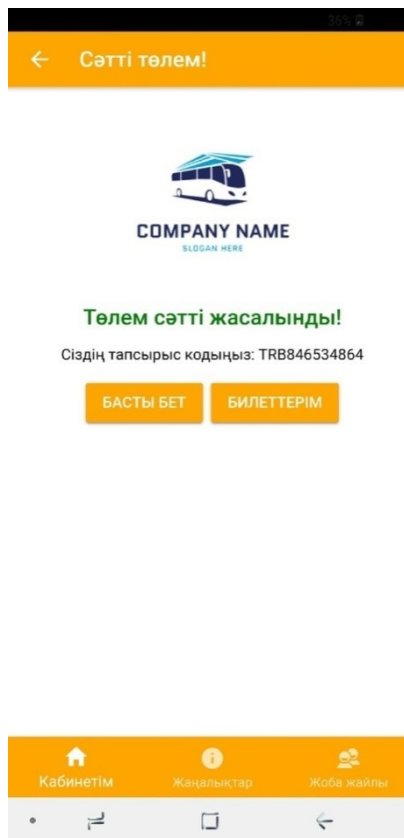


#### 4.8-сурет - Отыратын орындарды таңдау



#### 4.9-сурет - Сатылған билеттер

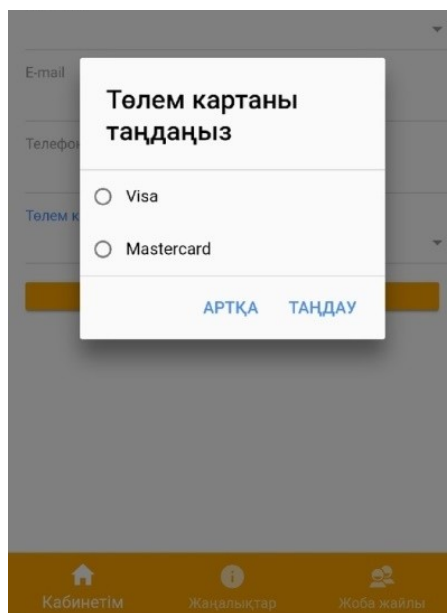
Содан кейын ғана барып (код) арқылы ғана автобекеттін сиздін қалаға кететін билетіңіз дайын.



#### 4.10-сурет - Билетті сатып алу

### 4.3 Менің карталарым

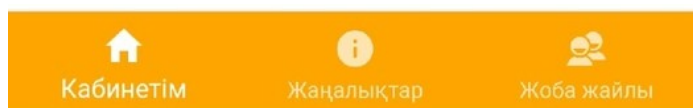
Ия әріне айта кететін жағдай менін жасаған бағдарламамда Менің карталарым бөлімінде карталар туралы барлық ақпараттар бар яғни карта арқылы қалай ақша төлеу керек екендігі



**4.11-сурет - Төлем жасау**

#### **4.4 Жаңалықтар**

Ал ендеше SATTI ZHOL”бағдарламасына қандай да жаңалықтар қандайда бір керекті ақпараттар болады болса бұл әріне Жаңалықтар постында барлық жаңалықтар болады.



**1.12-сурет Қосымша жаңалықтар**

## ҚОРЫТЫНДЫ

Қорытындылай келе, Интернетте әртүрлі әдебиеттер мен мәліметтерді пайдаланып, көптеген тақырыптар бойынша зерттеулер жүргізіп, талдағанымды айтааламын. Ғылым-техника ұшқан құстай дамыған 21-ғасырда білім беру жүйесін интернетсіз елестету мүмкін емес. Android қосымша құру технологиясы қазіргі кезде өзінің күрделілігі мен құпиялылығын есебінде кеңінен танымал, android қосымша құрудың әдістері өте көп, оның ішінде ешқандай шығынсыз ақ android қосымша құруға мүмкіндік беретін тәсілдер де жетерлік. Бұл дипломдық жобада android қосымша жобаланып, қосымша интерфейсі, оның функцияналдығы іске асырылды. Бағдарлама функционалдығына төмендегілер кіреді: қолданушыларды тіркеу, бағдарламаны басқару, деректер қорын басқарып деректерді құрылымдау және т.б. Қорыта келгенде, бұл Android платформасында қосымша жасалған бағдарлама басты міндеті тұтынушыға қажетті жағдайлар жасау, оларға интернет арқылы кешен туралы толық ақпарат алу, интернет арқылы кез келген уақытқа билеттірге тапсырыс беруге болады адамдар өз уақтын унемді пайдалана алатындай оларға көмегі тиетіныне сенідімін. Алматы қаласында болсын басқада қалаларда бұндай платформада жасалған бағдарлама бар екеніне көзім жеткені жоқ мен өз

бағдарламам үстінде жұмыс жасағанда мен алматы тағыда басқа қаласындағы автобекеттірмен(автовокзал) салыстырып отырып жасадым.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Халықаралық сауда және басқару журналы. [Электрондық құрылғы]. URL:[file:///C:/Users/Leyla/Downloads/ijrcm-1-vol-2\\_issue-10%20\(3\).pdf](file:///C:/Users/Leyla/Downloads/ijrcm-1-vol-2_issue-10%20(3).pdf) 08.12.2015
2. Wilson T, Wiebe J, Hoffman P. Фразеологиялық сезімталдықты талдау контекстік полярлықты тану. В: HTML / EMNLP материалдары. 2005., 04.12.2015
3. Орыс тілінде үрдістерді талдау жүйелерін бағалау. [Электронды ресурс]. URL: <https://aclweb.org/anthology/W/W13/W13-2403.pdf>, 26.12.2015
4. 10. Ұстамдылық талдауына кіріспе. [Электронды ресурс] .URL: <http://www.lct-master.org/files/MullenSentimentCourseSlides.pdf>, 10.01.2016
5. Webopedia. Веб-шабуылдар [Electronic resource]. URL: [http://www.webopedia.com/TERM/W/Web\\_Scraping.html](http://www.webopedia.com/TERM/W/Web_Scraping.html), 13.05.2016
6. Хартли Б. Мен қызықтыратын API қажеті жоқ: веб-шабуылдар мен пайда үшін. [Электронды ресурс]. URL: <https://blog.hartleybrody.com/web-scraping/>
7. Джурафский Д., Маннинг С. Шешімнің сегменті // Табиғи тілдерді өңдеу курсы. [Электронды ресурс]. URL: <https://class.coursera.org/nlp/lecture/5>, 19.03.2016
8. Табиғи тіл құралдар жиынтығы. [Электронды ресурс]. URL: <http://www.nltk.org/index.html>, 20.02.2016
9. Apache Open NLP. [Электронды ресурс]. URL: <https://opennlp.apache.org/>
10. Мехова Ю.А. Әлеуметтік медиа ағымдарының ішінде және оның әсерін талдау. [Electronic resource] .URL: <http://ir.uiowa.edu/cgi/viewcontent.cgi?Article=3090&context=ed>, 28.04.2016

## **В қосымшасы**

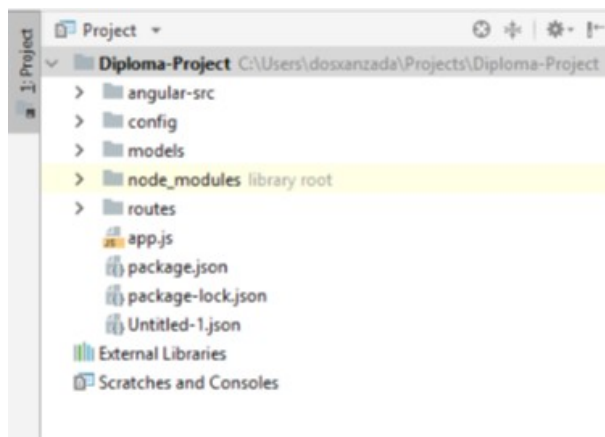
(міндетті)

«Android платформасында “SATTI ZHOL” қалааралық жолаушылар тасымалы мобильді қосымшасын әзірлеу» android - сайтының бағдарлама мәтіні

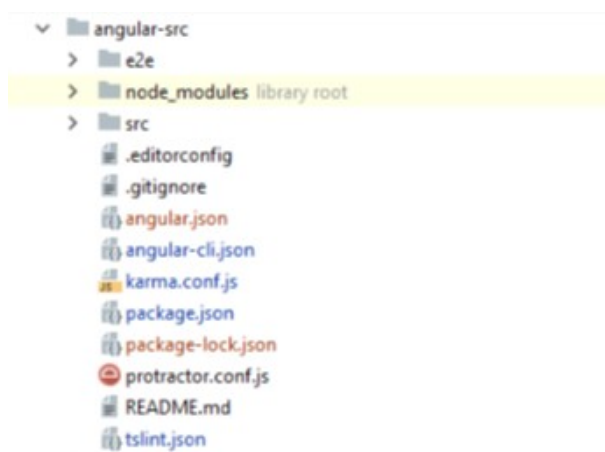
### **A.1 Жобаның каталогы**

Android платформасында -бағдарлама негізіндегі платформаның каталогы A.1-суретте көрсетілген. Жоба екі бөлім – серверлік бөлімнен және клиенттік бөлімнен құралған. Суретте angular-src – каталогында клиенттік бөлік орналасқан (A.2-сурет), ал басқа каталогтар серверлік бөлікті ұйымдастырады.





**А.1-сурет – Жобаның негізгі каталогы**



**А.2-сурет – Жобаның клиенттік бөлігін ұйымдастырушы каталог**

### **А Қосымшасының жалғасы**

Серверді ұйымдастырушы негізгі app.js файлының мәтіні:  
import { Component } from '@angular/core';  
import { AlertController, IonicPage, NavController, NavParams } from 'ionic-angular';  
import { AngularFireDatabase } from "angularfire2/database-deprecated";

```

import {HomePage} from "../home/home";
    //ДҚ қосылу
mongoose.connect(config.database);
//Қосылу жауабы
mongoose.connection.on('connected', () =>{
    console.log('Connected to database ' + config.database);
});
//Қателік туындаса
@IonicPage()
@Component({
    selector: 'page-authorization',
    templateUrl: 'authorization.html',
})
export class AuthorizationPage {

    username = "";
    password = "";

    //Порт нөмірі
    const port = 3000;
    app.use(cors());
    //Статикалық папка
    @IonicPage()
    @Component({
        selector: 'page-registration',
        templateUrl: 'registration.html',
    })
    export class RegistrationPage {

        //Passport Middleware
        app.use(passport.initialize());
        app.use(passport.session());
        require('./config/passport')(passport);
        app.use('/users', users);
        //Басты бетті қайтарушы
        app.get('/', (req, res) => {
            res.send('Invalid Endpoint');
        });
        //Сервер жұмысы басталды
        app.listen(port, () => {
            console.log('Server started on port ' + port);
        });
    }
}

```

Қалалар туралы мәлімет қызметін *passport.js* конфигурациялық файлының мәтіні:

```
<ion-label>Қай жерден</ion-label>
<ion-select [(ngModel)]="search.from" okText="Таңдау"
cancelText="Қайтару">
  <ion-option value="Нұр-Сұлтан">Нұр-Сұлтан</ion-option>
  <ion-option value="Алматы">Алматы</ion-option>
  <ion-option value="Талдықорған">Талдықорған</ion-option>
  <ion-option value="Көкшетау">Көкшетау</ion-option>
  <ion-option value="Шымкент">Шымкент</ion-option>
  <ion-option value="Тараз">Тараз</ion-option>
  <ion-option value="Өскемен">Өскемен</ion-option>
  <ion-option value="Павлодар">Павлодар</ion-option>
  <ion-option value="Ақтау">Ақтау</ion-option>
  <ion-option value="Ақтөбе">Ақтөбе</ion-option>
  <ion-option value="Атырау">Атырау</ion-option>
</ion-select>
</ion-item>
<ion-item>
  <ion-label>Қай жерге</ion-label>
  <ion-select [(ngModel)]="search.to" okText="Таңдау"
cancelText="Қайтару">
  <ion-option value="Нұр-Сұлтан">Нұр-Сұлтан</ion-option>
  <ion-option value="Алматы">Алматы</ion-option>
  <ion-option value="Талдықорған">Талдықорған</ion-option>
  <ion-option value="Көкшетау">Көкшетау</ion-option>
  <ion-option value="Шымкент">Шымкент</ion-option>
  <ion-option value="Тараз">Тараз</ion-option>
  <ion-option value="Өскемен">Өскемен</ion-option>
  <ion-option value="Павлодар">Павлодар</ion-option>
  <ion-option value="Ақтау">Ақтау</ion-option>
  <ion-option value="Ақтөбе">Ақтөбе</ion-option>
  <ion-option value="Атырау">Атырау</ion-option>
</ion-select>
</ion-item>
```

1. ДҚ ұйымдастырушы файл мәтіні:

```
module.exports = {
  database: 'mongodb://localhost:27017/diploma',
  secret: 'yoursecret'
}
```

## 2. Пайдаланушы моделінің мәтіні:

```
<ion-label>Күнді таңдаңыз</ion-label>
<ion-input type="date" [(ngModel)]="search.date"></ion-input>
</ion-item>
<ion-item>
  <button ion-button block color="myOrange"
(click)="searchBtn()">Автобустарды табу</button>
</ion-item>
</ion-list>
```

## А Қосымшасының жалғасы

```
email: {
  type: String,
  required: true
},
username: {
  type: String,
  required: true
},
password: {
  type: String,
  required: true
},
confirmPassword: {
  type: String,
  required: true
},
avatar: {
  type: String
}
});
const User = module.exports = mongoose.model('User', UserSchema);
module.exports.getUserById = function (id, callback) {
  User.findById(id, callback);
}
module.exports.getUserByUsername = function (username, callback) {
  const query = {username: username}
  User.findOne(query, callback);
}
module.exports.addUser = function (newUser, callback) {
```

```
bcrypt.genSalt(10, (err, salt) => {  
  bcrypt.hash(newUser.password, salt, (err, hash) => {  
    if (err) throw err;  
    newUser.password = hash;  
    newUser.save(callback);  
  });  
});  
}
```

## А Қосымшасының жалғасы

```
module.exports.comparePassword = function (candidatePassword, hash, callback){
  bcrypt.compare(candidatePassword, hash, (err, isMatch) => {
    if(err) throw err;
    callback(null, isMatch);
  });
}
```

### 3. Пайдаланушы моделінің қызметтерін ұйымдастырушы файл мәтіні:

```
<ion-label stacked>Логин</ion-label>
<ion-input [(ngModel)]="username" type="text"></ion-input>
</ion-item>
<ion-item>
  <ion-label stacked>Пароль</ion-label>
  <ion-input [(ngModel)]="password" type="text"></ion-input>
</ion-item>
<ion-item>
  <ion-label stacked>Есіміңіз</ion-label>
  <ion-input type="text" [(ngModel)]="firstname"></ion-input>
</ion-item>
<ion-item>
  <ion-label stacked>Тегіңіз</ion-label>
  <ion-input type="text" [(ngModel)]="lastname"></ion-input>
</ion-item>
<ion-item>
  <ion-label stacked>Туылған күн</ion-label>
  <ion-datetime [(ngModel)]="birthday"></ion-datetime>
</ion-item>
<ion-item>
  <ion-label stacked>Мекен-жай</ion-label>
  <ion-input type="text" [(ngModel)]="address"></ion-input>
</ion-item>
</ion-list>
```

## А Қосымшасының жалғасы

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { AngularFireDatabase } from "angularfire2/database-deprecated";
import { HelperService } from "../../shared/services/helper.service";
import { TicketModel } from "../../shared/models/ticket.model";

@IonicPage()
@Component({
  selector: 'page-find-ticket',
  templateUrl: 'find-ticket.html',
})
export class FindTicketPage {

  search: any = {
    from: 'Алматы',
    to: 'Нұр-Сұлтан',
    date: '2019-04-08'
  };

  constructor(public navCtrl: NavController, public NavParams: NavParams,
private firebase: AngularFireDatabase,
private helper: HelperService) {
}

  searchBtn() {
    this.firebase.list('/satti_tickets').subscribe((res: TicketModel[]) => {
      let result = res.filter((filter: TicketModel) => {
        return (filter.from == this.search.from) && (filter.to == this.search.to) &&
(filter.departureDate == this.search.date);
      });
      if (result.length == 0) {
        this.helper.presentToast('Бұл критерий бойынша билеттер табылмады!');
      } else {
        this.navCtrl.push('ListTicketPage', { result, params: this.search });
      }
    });
  }
}
```

}

## **А қосымшасы**

(міндетті)

### **Техникалық тапсырма**

#### **А.1.4 Клиенттердің тапсырыс берудың техникалық тапсырмасы**

Уақытты тиімді пайдалануда қазіргі қолданыстағы бағдарламалар көп септігін тигізуде. Бағдарлама қолданушылары кез келген уақытта, тиісті ақпараттарды алып, онлайн тапсырыс беріп, қанша адамдар қалайша сайтқа тіркелуі, бір күнде қанша автобустар жолға шығатындығы, бос орындар санын көрсетуін қамтамасыздандыру керек. Айта кететін болсақ әр түрлі елді мекендер арасындағы жолаушылар қатынасы тұрақты дамыды.

##### **А.1.4.1 Тапсырма**



Алматы қаласында Автовокзал (автобусбекеті) - жолаушылар мен тасымалдаушыларға қызмет көрсетуге арналған Android платформасында “SATTI ZHOL” қалааралық жолаушыларды тасымалдау мобильді қосымшасын әзірлеу арнайы бөлінген аумақта орналасқан кешенін қамтыйтын көлік инфрақұрылымының объектісі жасау билеттерге онлайн тапсырыс беру. Ол үшін дипломдық жұмыстың мақсатын және ығайлы жақтарын қарастыру. Қызметтің нәтижесі енгізуге және пайдалануға дайын бағдарламалық өнім болуы тиіс.

#### **А.1.4.2 Функционалдық талаптар**

Диаграммалармен жұмыс жасаған кезде былайша айтатын болсақ диаграмма бұл жасап жап жатқан мобильды қосымшасын әзірлеу бағдарламасынын жасаған кезде сол даргаммалар мен жұмыс жасап яғни тусіндіріп кетсем макет сияқты жасалады бағдарламаны жасау кезіндегі басты мақсат деугеде болады. Функциональды диаграммалар кезінде мен өзім бағдарламамен емес керекты бағдарламаларды алып солар арқылы диаграмманы жасап шықтым.

Диаграммалар жүйесін пайдалану талаптарын жинау үшін қолданылады. Сіздің талаптарыңызға байланысты бұл деректерді әр түрлі жолдармен пайдалануға болады. Оларды пайдаланудың бірнеше жолы бар:

### **А қосымшасының жалғасы**

#### **А.1.4.3 Сенімділікке қойылатын талаптар**

Енгізілетін ақпаратты бақылауды қамтамасыз етіңіз. Жүйемен жұмыс істеу кезінде пайдаланушының қате әрекеттерін қамтамасыз ету. Тапсырыс берген билеттерді қамтамасыз ету.

#### **А.1.4.4 Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар**

Жүйе IBM-үйлесімді дербес компьютерлерде жұмыс істеуі керек.  
Минималды конфигурация: процессордың түрі - Pentium және одан жоғары;

#### **А.1.4.5 Ақпараттық және бағдарламамен үйлесімділік талаптары**

Жүйе Win32 API интерфейсін іске асыратын Windows отбасының операциялық жүйелерінде жұмыс істеуі керек.

#### **А.1.4.6 Бағдарламалық құжаттамаға қойылатын талаптар**

Бағдарламаланған бағдарламалық модульдер өздігінен құжатталуы керек, яғни. Бағдарламалардың мәтіндері барлық қажетті түсініктемелерді қамтуы керек.

Бағдарламалық қамтамасыз ету жүйесінде қолданысқа қатысты анықтамалық ақпарат және пайдаланушыға ескертулер болуы керек.

