

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Сағатов Біржан Маратұлы

Транспорттық компанияға (такси) арналған ақпараттық жүйесін өңдеу

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

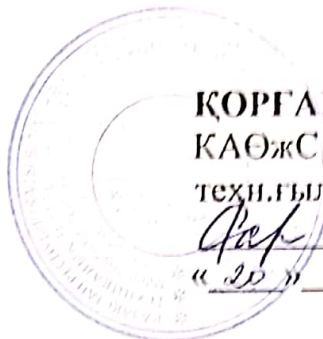
Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

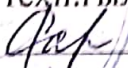
Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы



ҚОРҒАУҒА ЖІБЕРІЛДІ

КАӨЖС кафедрасы меңгерушісі

техн.ғыл.канд., ассон. профессор

 Р.Ж.Сатыбалдиева

« 20 » 05 2022 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Транспорттық компанияға (такси) арналған ақпараттық жүйесін өңдеу»

5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған

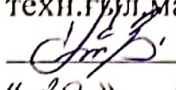
Сағатов Біржан Маратұлы

Резенцент

Ғылыми жетекші

техн.ғыл.кандидаты, "Ақпараттық жүйелер және киберқауіпсіздік" кафедрасының доценті, Алматы энергетика және байланыс университеті

 Тусупова Б. Б.
« 20 » 05 2022 ж.

техн.ғыл.магистрі, лектор
 Оган Аткелді,
« 20 » 05 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

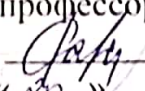
5B070300 – Ақпараттық жүйелер

БЕКІТЕМІН

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. кандидаты, ассоц.

профессор

 Р.Ж.Сатыбалдиева

« 20 » 05 2022 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы Сағатов Біржан Маратұлы

Тақырыбы: «Транспорттық компанияға (такси) арналған ақпараттық жүйесін өңдеу»

Университет Ректорының 2021 жылғы «24» желтоқсан №_489-П/Ө бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзімі «25» 05 2022 ж.

Дипломдық жұмыстың бастапқы мәліметтері:

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

1. Жобаға жалпы шолу;
2. Технологиялық құжаттама
3. Жоба диаграммалары түрінде ақпаратты жинау;
4. Бағдарламалық камтаманы құру;
5. Алынған мәліметтерге негізделген қорытынды.

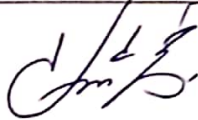
Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың 33 слайдтан тұратын презентациясы көрсетіледі.

Ұсынылған негізгі әдебиет 9 кітаптан тұрады.

Дипломдық жұмысты даярлау
КЕСТЕСІ

Бөлім атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімі	Ескерту
1. Дипломдық жобаның жоспарын құру	12.01.2022	
2. Негізгі бөлімді қарастыру	19.01.2022	
3. Жалпы жүйені жасау үшін қолданатын программалық камтамаларды дайындау	10.02.2022	
4. Жобалау бөлімін қарастыру	28.02.2022	
5. Қолданба жүйесін құру	20.03.2022	
6. Дипломдық жобаны қорытындылау	18.04.2022	

Дипломдық жұмыс бөлімдерінің кеңесшілері мен
норма бақылаушының аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдердің атауы	Кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған мерзімі	Қолы
Нормалық бақылаушы	Аристомбаева М.Т., техн.ғыл.магистр, лектор	19.05.2022	
Негізгі бөлім	Оған А., техн.ғыл.магистр, лектор	19.05.2022	

Ғылыми жетекшісі  Оған А.

Тапсырманы орындауға қабылдаған білім алушы _____ Сағатов Біржан
Маратұлы

Күні

«___» _____ 2021 ж

АҢДАТПА

Дипломдық жұмыстың тақырыбы - транспорттық компанияға (такси) арналған ақпараттық жүйесін өңдеу.

Бұл дипломдық жоба, такси компанияларға арналған ақпараттық жүйесін өңдеу болып табылады. Ақпараттық жүйе - қолданушыларға мобильді қосымша ретінде ұсынылатын арнайы агрегаторларды оңтайландырылған жүйе. Қазақстан нарығына басқа агрегаторлар енгізілгендіктен, бұл жүйе жаңа қызметтерге жобаларды сәтті іске қосуға және өз сервисін ұсынуға көмектеседі.

Дипломдық жұмысты өңдеу барысында Java бағдарламалық тілі, XML-кеңейтілетін белгілеу тілі және Firebase, Google Map API технологиялары қолданылды

АННОТАЦИЯ

Тема дипломной работы - разработка информационной системы для транспортной компании (такси).

Этот дипломный проект, является информационной системы для такси компаний, предлагаемая пользователям в качестве мобильного приложения. С внедрением на рынок Казахстана других агрегаторов, конкуренция между ними разрослась. Данная система поможет новым компаниям успешно запускать проекты и предоставлять свой сервис.

При разработке дипломного проекта были использованы язык программирование Java, XML-расширяемый язык разметки и технологии Android Studio, Firebase, Google Map API.

ANNOTATION

The topic of the thesis is the development of an information system for a transport company (taxi).

This diploma project is an information system for taxi companies offered to users as a mobile application. With the introduction of other aggregators to the market of Kazakhstan, the competition between them has grown. This system will help new companies to successfully launch projects and provide their service.

When developing the diploma project, the Java programming language, XML-extensible markup language and Android Studio, Firebase, Google Map API technologies were used.

МАЗМҰНЫ

Кіріспе	9
1 Аналитикалық бөлім	10
1.1 Қазақстандағы такси нарығы	10
1.2 Агрегаторлар	12
1.3 Қазақстандағы такси сервистері	13
1.4 Платформаны басқарудың негізгі алгоритмдеріне шолу	16
1.5 Тапсырыстарды бөлу алгоритмдері (matching)	16
1.6 Динамикалық баға белгілеу алгоритмдері (динамикалық баға)	18
1.7 Тапсырма қойылымы	19
2 Теориялық бөлім	20
2.1 Өңдеу технологиялары және баптау тілдері	20
3 Практикалық бөлім	24
3.1 Мобильді қосымшаның функционалы	24
3.2 Жүйенің жұмыс істеу блок схема және UML диаграммасы	29
Қорытынды	33
Пайдаланған әдебиеттер тізімі	34
А Қосымшасы	35
Б Қосымшасы	39

КІРІСПЕ

Жергілікті таксопарктер санының өсуі және таксидің тапсырысы бойынша халықаралық платформалардың қазақстандық нарығына кіру көліктің осы түрін қолжетімді етеді. Қазақстандық жолаушылар тасымалы нарығындағы соңғы жылдардағы айқын үрдіс-отандастарымыздың көбі күнделікті сапарлар үшін таксиді пайдаланады. Клиенттердің такси транспортының қолдануы, ең алдымен, нарықтағы бәсекелестікке байланысты қызмет құнын өзгердеті. Дипломдық жұмыстың негізгі мақсаты такси қызметін ұсынатын кәсіптерге арналған ақпараттық жүйе енгізу. Жұмыста пәндік аймаққа байланысты анализ жүргізіледі. Талаптар қойылып. Қазіргі уақытқа сай технологиялар мен қолданысқа енген жүйелерді зерттелінеді. Нәтижесінде берліген ақпаттар бойынша кәсіпке арналған ақпараттық жүйесін өндеуге қол жеткізіледі.

1 Аналитикалық бөлім

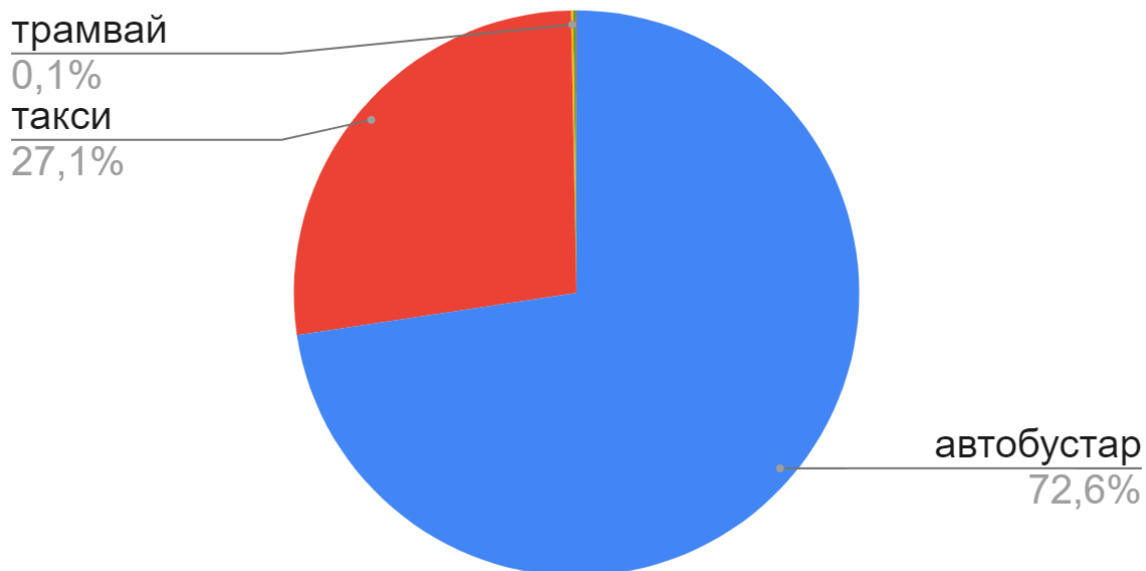
1.1 Қазақстандағы такси нарығы

Қазіргі уақытта Қазақстанда такси тәулігіне 900 мыңға жуық сапар жасайды, сапардың орташа чегі - 619 теңге. Такси нарығының көлемі жылына 120,8 млрд теңгені құрайды. Көлеңкелі нарықтың үлесі 95% - ға жетеді. Мұндай статистиканы Қазақстанның такси қызметтері республикалық қауымдастығының басшысы Медет Құрманов келтіреді.

Оның ұсыныстарының бірі-мемлекеттік ақпараттық жүйелерді құру. "Яғни, BIG Data негізінде такси саласында цифрлық реттеу құрылуы тиіс. Кейіннен қалалық мониторинг орталықтары сапаны бақылауы тиіс (такси жүргізушілері көрсеткен қызметтер – Ред.). Мысалы, Алматыда қала өмірінің түрлі салалары бойынша қалалықтардың өтініштерін Open Almaty қабылдайды. Олар такси жұмысына қатысты шағымдар мен ұсыныстарды қабылдай алады", - дейді Медет Құрманов.

Қазақстан бойынша такси транспортымен жолаушыларды тасымалдаудың үлесі аз емес. 2017-2018 жыл арасындағы тасымалдаудың жалпы санының 27,1% - ы такси транспорт түрі құрайды (1-сурет), (1-кесте). [1]

2018 жылғы ҚР-дағы такси транспортының жолаушыларды тасымалдау жиыны



Сурет 1. 2018 жылғы ҚР-дағы транспорттарының жолаушы тасымалдау жиыны

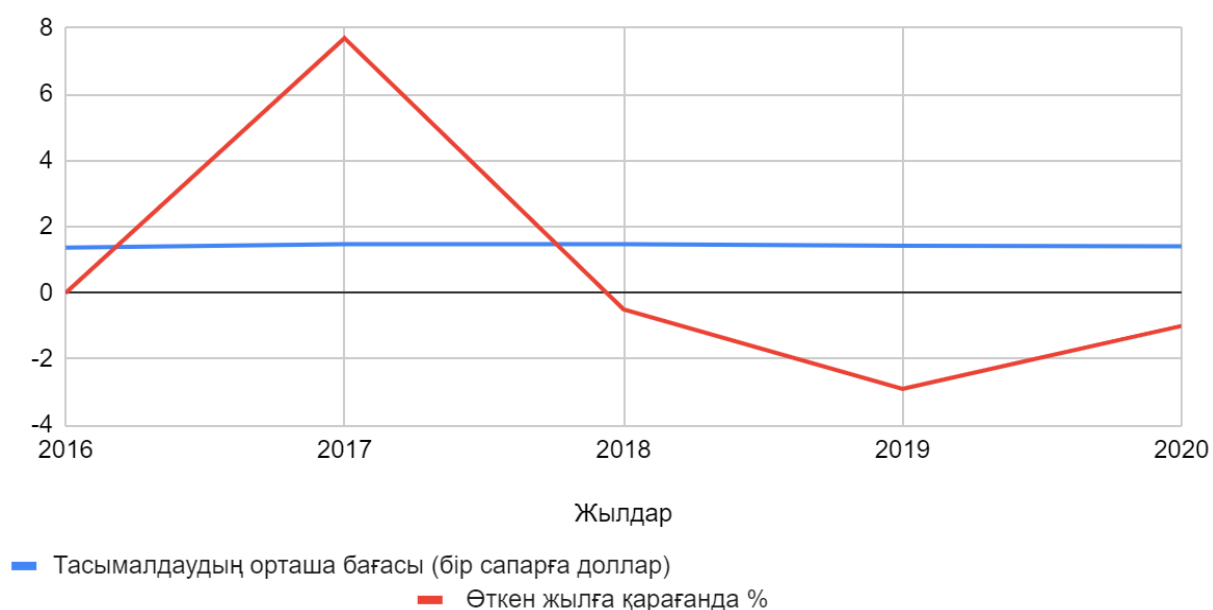
	2018	2017	Бір жылдағы өсім	Үлесі
жиыны	18 882,5	18,573,2	1,7%	100%
автобустар	13 713,5	14952,4	-8,3	72,6%
такси	5 110,1	3562,3	43,4%	27,1%
трамвай	23,8	25,8	-7,7%	0,1%
троллейбус	22,3	20,8	6,9%	0,1%
метрополитен және көліктің басқа түрлері	12,7	11,9	6,9%	0,1%

Кесте 1. 2018 жылдың қаңтар-қазан айында жолаушыларды тасымалдау саны

Қазақстанда такси тасымалдау нарығының көлемі ақшалай түрде 120 млрд теңге, 195 млн жол жүріс құрайды. Тасымалдардың 70% - ы республикалық маңызы бар 3 қалаға тиесілі.

Такси қызметтерінің орташа бағасы нарық айналымының жасалған сапарлар санына қатынасы ретінде анықталады (2-сурет), (2-кесте).[1][2]

Тасымалдаудың орташа бағасы (бір сапарға доллар) и өткен жылға қарағанда %



Сурет 2. Тасымалдаудың орташа бағасы

Көрсеткіш	2016	2017	2018	2019	2020
Тасымалдаудың орташа бағасы (бір сапарға доллар)	1,37	1,47	1,47	1,42	1,41
өткен жылға қарағанда %	-	7,7	-0,5	-2,9	-1,0

1 – кесте-Тасымалдаудың орташа бағасы

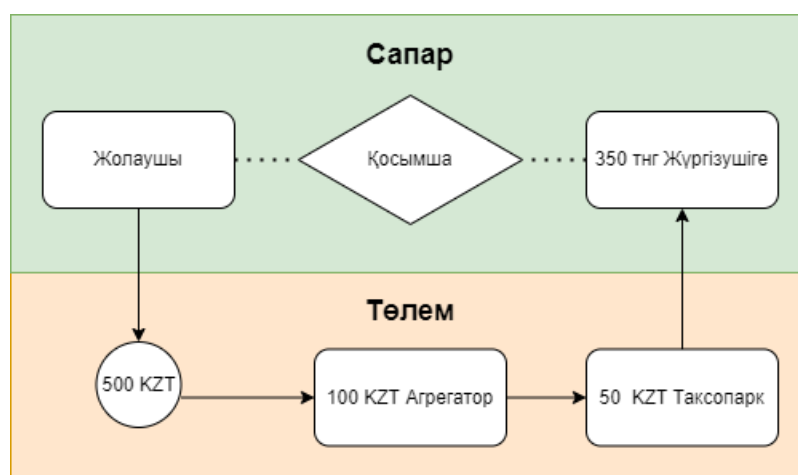
1.2 Агрегаторлар

Интернеттің дамуы такси агрегаторларының пайда болуына өз әсерін тигізді. Агрегаторлар таксопарктерді біріктіре бастады. Uber 2009 жылы бірінші болып пайда болды, одан кейін 2011 жылы Яндекс таксиі мен Гетт.

Шын мәнінде, агрегаторлар-жүргізушіні жолаушымен байланыстыратын бағдарламаларды жасаушылар. Кең жарнамалық мүмкіндіктердің арқасында әлеуетті жолаушылар смартфонға арналған қосымшаны жүктеп алып, ол арқылы таксиге тапсырыс берсеңіз, ол арзан, ыңғайлы және қауіпсіз болатынын біледі.

Агрегаторлардың жалпы жұмыс принципі (3 – сурет):

- Тапсырыстар қайдан келеді. Жолаушы мобильді қосымша арқылы жүргізушімен байланысады.
- Жүргізушілерді кім қосады. Агрегаторлар: Яндекс такси, Uber немесе InDriver заңды тұлғалармен және жеке кәсіпкерлермен, ал олар өз кезегінде жүргізушілермен жұмыс істейді.
- Комиссия. Агрегаторлар мен таксопарктер делдалдық үшін тапсырыстың пайызын алады. Мысалы, Uber 20%, ал таксопарк 10% алады.[3]



Сурет 3. Агрегаторлардың жұмыс істеу қағидасы

Неліктен танымал?

- Агрегаторлар сапа мен қауіпсіздікті қамтамасыз ететін автомобиль мен жүргізушіге жоғары талаптар қояды. Жолаушыға ыңғайлы.
- Көбінесе сапардың құны алдын-ала белгілі. Сіз кез-келген көлікке тапсырыс бере аласыз: эконом және бизнес-класс, балалар орындығы немесе минивэн. Бұл жолаушыларға да ұнайды.
- Агрегаторларға қосылған машиналардың көптігіне байланысты таксиді күтудің қажеті жоқ. Мысалы, Мәскеуде өтініш беруден бастап өтініш беруге дейін 10 минуттан аспайды. [3]

1.3 Қазақстандағы такси сервистері

2017 жылы Қазақстанда таксимен жүрудің орташа бағасы 7,7% - ға өсті: 1,37 доллардан 1,47 долларға дейін сапар. Өсім орташа доллар бағасының айтарлықтай төмендеуінен кейін 2016 сапарлары қалпына келді. "Uber", "Яндекс Такси" (4-сурет) және басқа агрегаторлар Қазақстан нарығына шыққаннан кейін ол 2 есеге жуық – бір сапар үшін 2,29 доллардан 1,37 долларға дейін төмендеді. Агрегаторлар клиенттерді бір-бірінен тартып, белсенді түрде демпинг жасады. 2016 жылы таксиге долларлық бағаның құлдырауы ұлттық валютаның девальвациясына да ықпал етті.

Қазақстанда жұмыс істейтін жиі қолданылатын сервистер ЯндексGo(Taxi), Uber және InDriver.

Яндекс.Такси қызметі Қазақстан нарығына шығып, 28 шілдеде алматылық қолданушыларға қолжетімді болды.

Қазақстанда іске қосу үшін компания Яндекс.Такси қосымшасы мен веб-сайтының интерфейсін қазақ тіліне аударды. Орыс, ағылшын және армян тілдері де бар.

Жол жүру құны алғашқы екі шақырымға 100 теңгені (РФ Орталық банкінің 28 шілдедегі бағамы бойынша 18,73 рубль) және келесі екі шақырымға 50 теңгені (9,4 рубль) құрайды.

Қызмет өкілдерінің айтуынша, Алматыда Яндекс.Таксиге тоғыз қалалық такси компаниясының 150-ден астам көлігі қосылған. Іске қосылғаннан кейін бірнеше ай ішінде компания такси жүргізушілерінен комиссия алмақ емес.

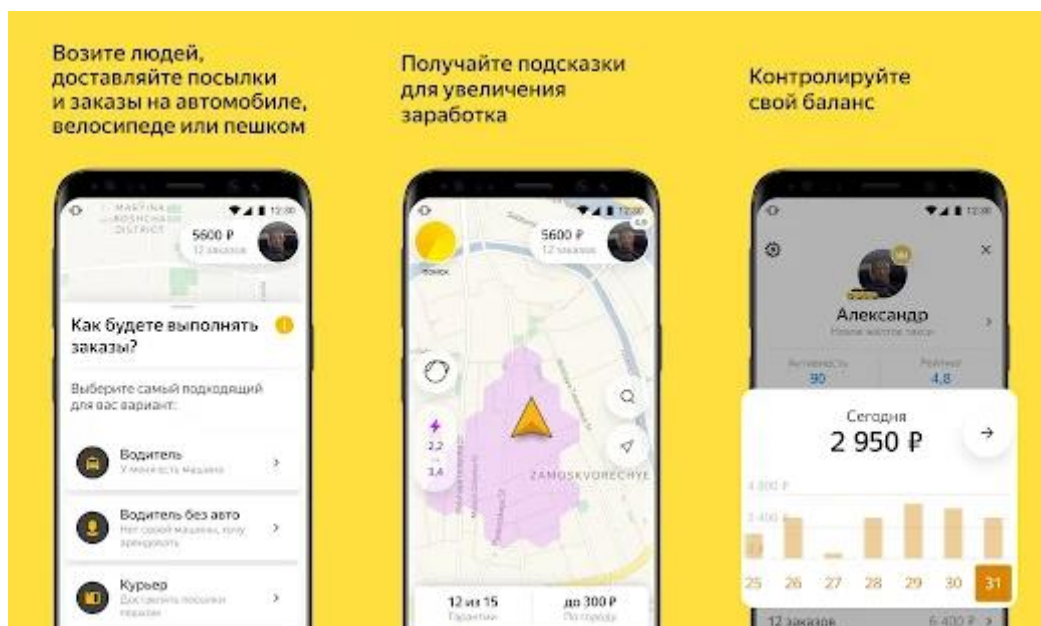
Қазір «Яндекс.Такси» Ресейдің 20-дан астам қаласында, сонымен қатар Минск, Ереван және Алматыда жұмыс істейді. Компанияның мәліметінше, қызметке 600 такси компаниясының 70 мыңнан астам көлігі мен диспетчерлік пункт қосылған.

2016 жылдың 10 маусымында Uber Қазақстанның астанасы Астанада UberX бюджеттік қызметін іске қосты. Шілде айында Uber Алматыда

жұмысын бастады. Қазақстан ТМД-да Ресей, Беларусь және Әзірбайжаннан кейінгі төртінші мемлекет болды.

Яндекс такси сервисы 2022 жылы Яндекс GO қосымшасына енді. Яғни, тапсырушы тағамдарды және такси қызметтерін бір қосымшаның ішінде жасай алады. [4]

Ал жүргізушілер үшін тапсырыстар келетін жеке қосымша бар. Осы жерде айта кететін жайт, қазіргі уақытта Uber мен Яндекс такси ынтымақты. Және барлық Uber және Yandex Taxi тапсырыстары бір Yandex PRO қосымшасында келеді.



Сурет 4. Яндекс такси қосымшасы

Сервис жолаушыларға жайлылығына қарай бірнеше класс санаттарына бөлген.[5]

Яндекс таксиінде келесі класстар санаттары бар:

- 1) Эконом;
- 2) Комфорт: Skoda Octavia, Hyundai Elantra, Hyundai Elantra;
- 3) Комфорт+: Toyota Camry, Kia Optima, Hyundai Sonata;
- 4) Business: Mercedes-Benz E-class, BMW 5, Audi A6;
- 5) Premier: Mercedes-Benz S-class, BMW 7, Audi A8;
- 6) Elite: Mercedes-Maybach S-class.

Және сервисіте аккаунт түрін таңдау мүмкіндігі бар. Аккаунттың 3 түрі бар, олар негізгі, отбасылық, бизнес. Негізгі аккаунт бойынша әдеттегі тапсырыс схемасы бойынша тапсырыс бересіз. Отбасылық аккаунт бір карта бойынша үш қолданушыға төлем жасауға мүмкіндік береді. Бизнес-аккаунт бойынша кәсіпкерлер мен қызметкерлер такси және тапсырыстар қызметін қағаз шарттарсыз және күрделі келісімдерсіз пайдалана алады.

Сондай-ақ, қосымшадағы клиенттер тапсырыс тарихын (мекен-жайы, күні және уақытты) көре алады.

Жүргізушілерді тіркеу принципі келесі реттілікке ие:

- 1) Жүргізуші телефон нөмірімен немесе поштамен тіркеледі;
- 2) Пароль қояды;
- 3) Таксопаркті тіркейді;
- 4) класс санатына байланысты оқыту және автомобильді тіркеу;
- 5) деректерді енгізеді және растайды (куәлік, жүргізуші куәлігі, машина нөмірі және т. б.).[5]

Соңғы онжылдықта көлік шешімдері индустриясы жаңа тұжырымдамалық идеялар мен технологиялар, мысалы, онлайн-Жолаушылар тасымалы платформалары, өзін-өзі басқаратын автомобильдер мен электр қозғалтқышы бар автомобильдерді дамыту арқылы қарқынды дамып келеді. Көптеген компаниялар мен зертханалар бір уақытта жұмыс істейтін осы технологиялардың синергиясы жолаушылар тасымалының нақты құнын бірнеше онжылдықта 10 есе төмендетуде үлкен серпіліс береді.

Қазіргі уақытта технологияның осы тізіміндегі ең қарқынды өсуді онлайн жолаушылар платформалары көрсетеді. Мысалы, Uber 10 жыл ішінде бүкіл әлем бойынша 80-нен астам елде және 700 қалада 10 миллиардтан астам сапар жасады. Мұндай онлайн-тасымалдаудың әлемдік нарығы 2030 жылға қарай 285 миллиард долларға жетуге уәде береді. Сондықтан жолаушылар мен жүргізушілердің екіжақты нарығын динамикалық түрде басқаратын осындай платформалардың тиімділігі үлкен практикалық маңызы бар екендігі таңқаларлық емес.[6]

inDriver-орталықтандырылмаған модель бойынша жұмыс істейтін халықаралық жолаушылар көлігі қызметі. Сапардың барлық шарттары жүргізушілер мен жолаушылар арасындағы келісім нәтижесінде анықталады

InDriver мобильді қосымшасы Android Және iOS платформалары үшін қол жетімді. Сервистің әлемнің 42 елінде 100 миллионнан астам пайдаланушысы бар (1.5-сурет).[6]

2017-2018 жылғы қысқы маусымда сервистің қамтуы Ресей мен ТМД елдерінің 70-тен 120-ға дейін қалаларына ұлғайды, ал 2018 жылы жасалған сапарлардың жалпы айналымы 300 млн долларға баламасынан асты (5-сурет).[6]



Сурет 5. InDriver қосымшасы

1.4 Платформаны басқарудың негізгі алгоритмдеріне шолу

Эмпирикалық зерттеулер деректерді өңдеудің, маршруттаудың, баға белгілеудің және тапсырыс берудің автоматтандырылған алгоритмдері онлайн платформаларға жүргізушілердің жұмыс уақытын неғұрлым көп жоюға және классикалық такси қызметімен салыстырғанда жолаушылар үшін көлік күту уақытын аз күтуге мүмкіндік береді. Сонымен қатар, жүйенің осы екі негізгі сипаттамасы (жүргізушілердің уақытын жою және жолаушының күту уақыты) қызметтің сенімділігі мен тұрақтылығымен тығыз байланысты: сұраныстың кенеттен жергілікті өршуі (мысалы, үлкен концерттің соңында немесе жаңа жыл қарсаңында) екі метрикуаны да едәуір нашарлатуы мүмкін, осылайша қызметті пайдалану нарықтың екі жағы үшін де тартымды емес. Бұл жоғары сұранысқа ие аймақтағы желідегі жүргізушілер тапсырыстардың жалпы санының аз бөлігін тез алады, ал қалған тапсырыстарды алыс аудандардан жүргізушілер тағайындайды. Бұл автомобильді жеткізу уақытын арттырады, ол көбінесе жүргізушіге төленбейді (бұл уақыт бірлігінде оның табысын азайтады) және сонымен бірге жолаушыға жағымсыз әсер қалдырады. Осылайша, платформаны пайдаланатын екі тарап оны аз қолдана бастайды. Осыған байланысты екі метрика да нашарлай бастайды, осылайша платформаның төмен спиралын нөлдік тиімділік бағытында бұрайды. Ағылшын әдебиетінде бұл теріс құбылыс (Wild Goose Chase (WGC)) деп аталады, оның аудармасы "жабайы қазды құдалау".

Платформаның тұрақтылығы мен өнімділігін арттыруға бағытталған екі негізгі технология — тапсырыстарды бөлу және динамикалық баға алгоритмдері. Бірінші технология диспетчерлік шешімдерді бақылайды, ал нақты уақыт режимінде динамикалық баға жолаушылар тасымалына сұраныс пен ұсыныстың өте құбылмалы арақатынасын теңестіреді. Динамикалық баға жүйенің жұмыс қабілеттілігін сақтау, автомобильді жеткізуді күту уақытын азайту және жоғары сұраныс кезеңінде жүргізушілер санын көбейту үшін өте қажет. Сонымен қатар, эмпирикалық және теориялық зерттеулер динамикалық баға WGC патологиялық қауіпті әсерінің ауқымын азайтуға мүмкіндік беретінін көрсетеді.

1.5 Тапсырыстарды бөлу алгоритмдері (matching)

Тапсырыс бойынша жүргізушіні тағайындаудың қарапайым басқару алгоритмі - ең жақын жүргізушіні тағайындау протоколы (first-dispatch protocol). Оның қарапайымдылығы мен жақсы практикалық көрсеткіштеріне қарамастан, бұл алгоритм жиі кездесетін жағдайлардың көпшілігінде тиімсіз екенін көрсету оңай. Біріншіден, ол жүргізушіні жаңа тапсырысқа жақын

жерде сапарды аяқтауға жақын болуы мүмкін жүргізушілерді елемей, тапсырыс кезінде бос тұрған жүргізушілердің кіші тобынан ғана таңдайды. Екіншіден, бұл қарапайым алгоритм белгілі бір уақыт аралығында жүйе туралы ақпаратты ғана ескереді, ал көбінесе платформада Тапсырыс ағынымен және жақын болашақта жүргізушілердің кеңістіктік таралуымен не болатыны туралы нақты ақпарат болуы мүмкін. Әдебиетте алгоритмнің сапасын жақсарту үшін осындай ақпарат ретінде практикалық рецепттер беретін ұқсас тапсырмалар класы "k серверлерінің мәселесі" (the K-server problem) деп аталады (6-сурет).[3]

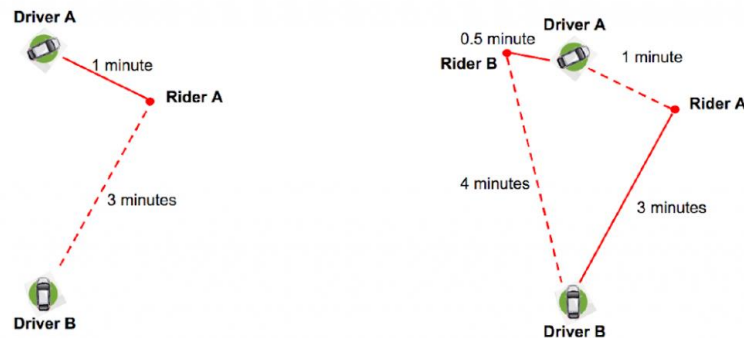


Figure 4 An illustration of efficiency loss in myopic matching algorithms.

Сурет 6. Тапсырыс бөлу алгоритімі

Диспечерлік алгоритмдерінің тағы бір танымал тобы қысқа уақыт аралығында саяхаттарға Тапсырыс тобын біріктіру және жұптық оңтайландыру мәселесін шешу идеясына негізделген. Басқаша айтқанда, әрбір жеке тапсырыс үшін автомобильді жедел және дәйекті тағайындаудың орнына, жүйе кіріс тапсырыстары туралы ақпаратты жинайды және жинақталған тапсырыстарды желіде бар жүргізушілер арасында белгілі бір жиілікпен таратады. Егер кейбір тапсырыстар тағайындалған жүргізушісіз қалса, онда олар жүйеде қалады және келесі уақыт қадамын бөлу міндетіне қатысады. Әр қадамда шешілетін оңтайландыру мәселесінің мақсатты функциясы генерацияланатын диспетчерлік тағайындаулардың сапасын сипаттайтын өлшемдердің кең спектрін қамтуы мүмкін: жолаушының автокөлікті күту уақыты, тапсырыс пен тағайындалған жүргізуші арасындағы қашықтық, жолаушының немесе жүргізушінің Тапсырысты жою ықтималдығы және т.б.

Іс жүзінде диспечерлік алгоритмдері әлдеқайда күрделі болып көрінеді, өйткені олар қосымшаның интерфейсында бір уақытта ұсынылған әртүрлі өнімдердің көптеген ерекшеліктерін ескеруі керек. Мысалы, платформада тіркелген автомобильдер әртүрлі жайлылық класстары мен сыйымдылығы әртүрлі болуы мүмкін. Интернеттегі платформалардың кейбір өнімдері, егер олардың маршруттары жақын болса, әр түрлі жолаушылар (UberPool, Lyft Line) бір автомобильді бір уақытта пайдалануды тұтады. Сонымен қатар,

диспетчерлік шешімдер көбінесе қызмет көрсету аймақтары мен оларға келіп түсетін тапсырыстардың бағыттары бойынша жүргізушілердің қалауын ескеруі керек. Осылайша, нақты уақыт режимінде шешілуі қажет диспетчерлік шешімдердің тиімділігін арттыруға бағытталған оңтайландыру міндеттерінің спектрі үнемі жаңа күрделі тұжырымдамалармен толықтырылып отырады.[3]

1.6 Динамикалық баға белгілеу алгоритмдері (динамикалық баға)

Кеңістіктегі және уақыттағы сұраныс пен ұсыныстың тепе-теңдігін бақылау үшін онлайн-платформалар динамикалық баға алгоритмдерін қолданады, олар нақты уақыт режимінде базалық тарифті арттырады, егер жолаушылар тарапынан түсетін тапсырыстар саны еркін жүргізушілер санынан едәуір көп болса. Платформаның тұрақты жұмысын қамтамасыз ету үшін динамикалық баға белгілеудің пайдасы жүйеге рекордтық жоғары жүктемелермен байланысты көптеген теориялық модельдермен, тәжірибелермен және эмпирикалық бақылаулармен расталады. Мұндай жүктемелер болжамды және өте көп емес себептерге байланысты болуы мүмкін: қолайсыз ауа-райы жағдайлары, бұқаралық шаралар, қоғамдық көлік жүйесінің дұрыс жұмыс істемеуі және т.б. Баға белгілеу алгоритмі дұрыс жұмыс істемеген жағдайда, жолаушылар тарапынан сұраныстар санының күрт артуымен (немесе қолжетімді автомобильдер санының күрт төмендеуімен) нәтижесінде автомобиль тағайындалатын жолаушылардың өте төмен үлесін және оны берудің қанағаттанарлықсыз жоғары уақытын байқауға болады. Онлайн платформа үшін динамикалық баға белгілеудің басты рөлі-кез-келген қолданушыға кез-келген жерде және кез-келген уақытта такси шақыруға мүмкіндік беру. Ұсынылған тариф әдеттегіден жоғары болса да, бұл қазіргі уақытта қол жетімді көліктер жоқ екендігі туралы платформаның пайдаланушысын (шұғыл түрде машина қажет болуы мүмкін) орналастырудан гөрі қолайлы нұсқа болады.

Динамикалық бағаны модельдеудің танымал әдістеріне жүйенің стационарлық күйін сипаттайтын экономикалық модельдер (тұрақты күй модельдері), динамикалық бағдарламалау модельдері, регрессиялық талдау, сонымен қатар желілік эффектілерді сипаттайтын оңтайландыру модельдері (желіні оңтайландыру) жатады. Соңғы экономистердің зерттеулері (Кастилло, 2017) көрсеткендей, тарифтің динамикалық өсуі платформаға жоғарыда айтқан WGC теріс әсер ету аймағына енбеуге мүмкіндік береді.

Динамикалық баға белгілеудің объективті кемшіліктері бар. Біріншіден, таксиге тапсырыс берген кезде жолаушылар көретін сапардың соңғы бағасы сұраныс пен ұсыныстың құбылмалылығына байланысты айтарлықтай өзгеруі мүмкін, осылайша сол маршруттағы тарифтің белгісіз өзгеруін арттырады. Екінші жағынан, онлайн-платформалардың

жүргізушілері көбінесе қаланың өсу коэффициенті белсенді аудандар туралы ақпаратқа қол жеткізе алады. Алайда, осы коэффициенттің жоғары құбылмалылығына байланысты, жүргізуші жоғары баға аймағына ауысқан кезде, тариф базалық мәндерге оралуы мүмкін. Сонымен қатар, тарифті алгоритммен автоматты түрде көтеру жүргізушілерді жергілікті нарықта тапсырыс беру үшін қол жетімді автомобильдердің жетіспеушілігі жағдайларын жасанды түрде құруға және жол жүру коэффициенттерін арттыруға итермелеуі мүмкін. Әрине, жүргізушілердің мұндай келісілген мінез-құлқын тапсырыс туралы көптеген мәліметтерді өңдейтін платформалар үшін анықтау және қажетті қорғаныс әрекеттерін қабылдау оңай, бірақ жасанды түрде көтерілген жолаушылар үшін мұндай тәжірибе қанағаттанарлықсыз болуы мүмкін.

1.7 Тапсырма қойылымы

Транспорттық компанияға арналған ақпараттық жүйе құрастыру арқылы қарапайым адамдардың үйден шықпай ақ таксиға тапсырыс беру мүмкіндігі.

Алға қойылған мақсаттарға жету талаптары осындай болмақ:

- ақпараттық технологиялар мен құралдарды таңдау;
- деректер қорын құрастыру;
- ақпараттық жүйені құрастыру;
- ақпараттық жүйені тестілеу.

2. Теориялық бөлім

2.1 Жүйенің жұмыс істеу блок схема және UML диаграммасы

ISO / IEC 19505-2: 2012 Бірыңғай модельдеу тілін (UML) анықтайды. UML диаграмманың мақсаты-жүйелік архитекторларға, бағдарламашы инженерлеріне және бағдарламалық жасақтама әзірлеушілерге бағдарламалық жүйелерді талдау, жобалау және іске асыру құралдарын ұсыну, сондай-ақ бизнесті және ұқсас процестерді модельдеу.

UML диаграмма арқылы программаның жұмыс істеу моделін көрсету үшін келесі диаграмма түрлерін ажыратуға болады:

- Прецеденттер диаграммасы (use-case diagram);
- Сынып диаграммасы (класс диаграммасы);
- Әрекет диаграммасы (Activity diagram);
- Реттілік диаграммасы (sequence diagram);
- Орналастыру диаграммасы (Deployment diagram);
- Ынтымақтастық диаграммасы (Collaboration diagram);
- Нысан диаграммасы (Object diagram);
- Мемлекеттік Диаграмма (Statechart diagram).

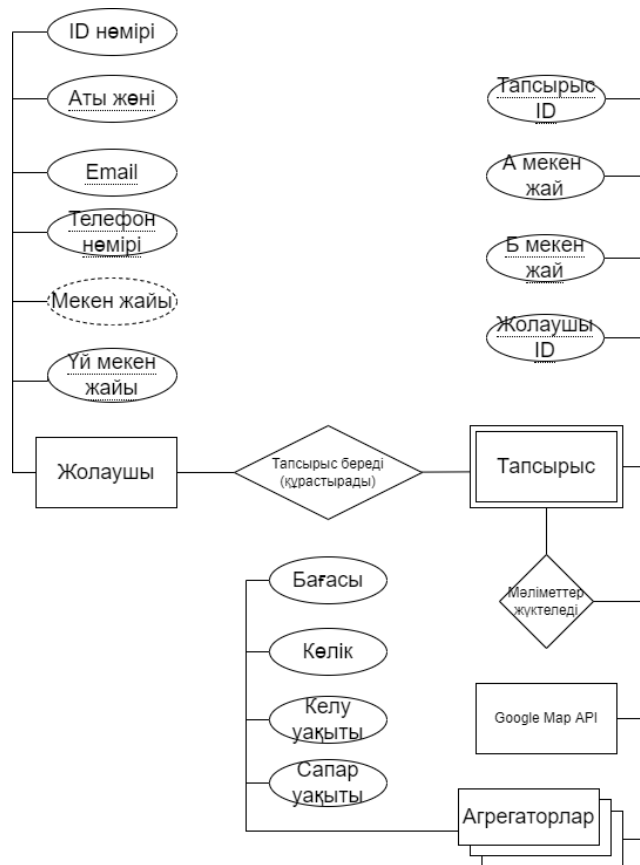
Use Case диаграмма-жүйенің прецеденттерімен(белгілігі оқиғамен) жұмыс істейтін объекттерді және өзара прецеденттердің қандайда бір байланысын көрсететін модельдеу диаграммасы.

"Мәні-байланыс" (немесе ERD) диаграммалары — қарапайым және күрделі дерекқорларды қоса алғанда, кез келген жүйені модельдеу процесінің ажырамас құрамдас бөлігі, алайда оларда қолданылатын фигуралар мен белгілеу әдістері кез келген адамды оңай адастыруы мүмкін. Бұл нұсқаулық сізге ER диаграммаларын белгілеу бойынша нағыз маман болуға және өз деректер базаңызды модельдеуге сенімді болуға көмектеседі!

Тұжырымдамалық деректер модельдері модельге не кіруі керек екендігі туралы жалпы түсінік береді. Тұжырымдамалық ER диаграммаларын логикалық деректер модельдерінің негізі ретінде алуға болады. Оларды интеграцияның негізін құра отырып, әртүрлі ER модельдері арасында қауымдастық қатынастарын құру үшін де қолдануға болады. Төмендегі барлық таңбаларды UML үшін "мәні-байланыс" және Lucidchart-тағы" мәні-байланыс "моделі бойынша фигуралар кітапханаларынан табуға болады.

ERD-нысан таңбалары

"Субъектілер" ұғымы маңызды ақпаратты беретін объектілерді немесе ұғымдарды білдіреді. Грамматика тұрғысынан олар әдетте зат есімдермен белгіленеді, мысалы, " тауар", " клиент", " мекеме "немесе"промоакция". Төменде 7-суретте біздің ақпараттық жүйенің ER диаграммасы бейнеленген.[9]



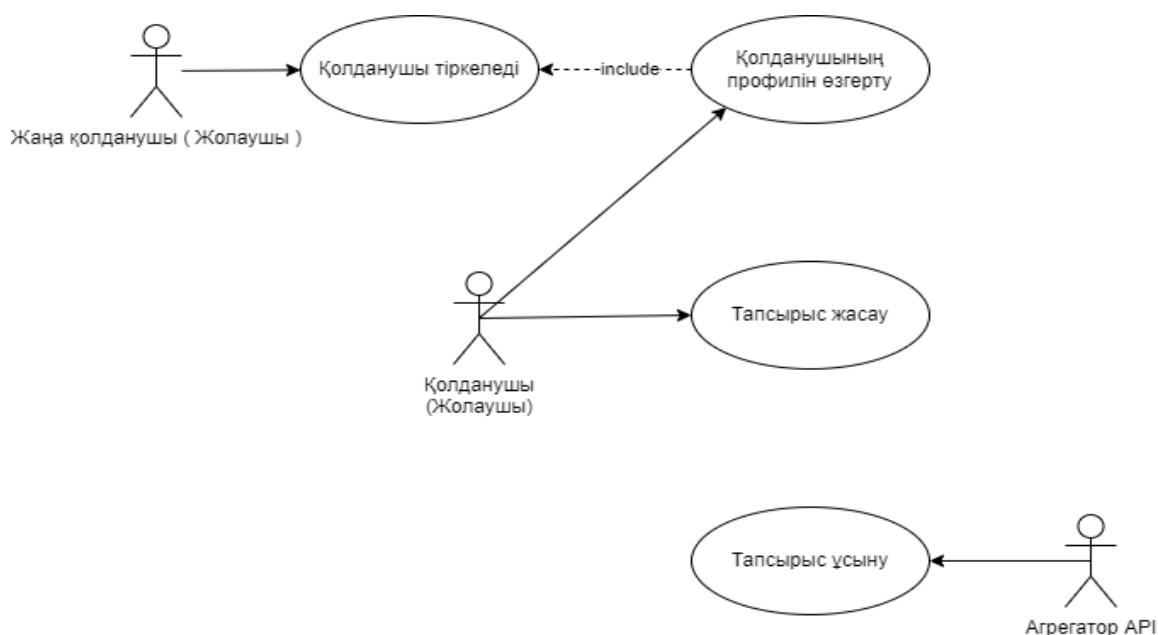
Сурет 7. ER диаграмма

Диаграмманың негізгі элементтері-қатысушы (actor) және прецедент (нұсқа).

Қатысушы-бұл прецеденттермен немесе субъектілермен (жүйе, ішкі жүйе немесе сынып) өзара әрекеттесу кезінде орындалатын логикалық байланысты рөлдердің жиынтығы. Қатысушы субъектіден тыс нәрсені білдіретін адам немесе басқа жүйе, ішкі жүйе немесе сынып бола алады. Графикалық түрде қатысушы "адам" ретінде бейнеленген.

Прецедент (use case) - қатысушы байқайтын нәтижеге әкелетін жүйе орындайтын көптеген жүйелі оқиғалардың (нұсқаларды қоса алғанда) сипаттамасы. Прецедент қатысушылар мен жүйенің өзара әрекеттесуін сипаттайтын субъектінің әрекетін білдіреді. Прецедент белгілі бір нәтижеге "қалай" қол жеткізілгенін көрсетпейді, тек "не" дәл орындалады. Прецеденттер өте қарапайым түрде көрсетіледі - эллипс түрінде, оның ішінде оның атауы көрсетілген.

Төменде 8-сурет бойынша жүйенің прецедент диаграммасы көрсетілген.[8]



Сурет 8. Use case диаграммасы

Жаңа қолданушы – жүйеге өзі жайлы ақпарат енгізеді. Кейін ол қолданушы болып өзі жайлы ақпаратты өзгерте алады.

Қолданушы – жаңа тапсырыстарды жасай алады. Және тапсырысты жоя алады.[8]

Агрегатор – берілген ақпараттар бойынша тапсырыс ұсына алады. Егер жолаушыға ұсынылған тапсырыстар қанағаттандырса ол сол тапсырысты таңдайды.[8]

Қызмет диаграммасы (ағылш. activity diagram) – күйлері күй диаграммасында сипатталған әрекеттерді көрсететін UML диаграммасы. Астында қызметпен (ағыл. activity) бағыныңқы элементтерді — ендірілген қызмет түрлерін және ағылш. бір түйіннің шығуынан екінші түйіннің кірісіне өтетін ағындармен өзара байланысты әрекет.[8]

Күй диаграммалары 9-сурет бизнес-процестерді, технологиялық процестерді, дәйекті және параллель есептеулерді модельдеуде қолданылады.[8]

Тапсырыс алгоритмі



Сурет 9. Қызмет диаграммасы

3 Практикалық бөлім

3.1 Өңдеу технологиялары және баптау тілдері

Соңғы уақытта ақпараттық технологиялардың қарқынды дамуы планшеттер, смартфондар, нетбуктар және басқа да гаджеттер сияқты көптеген жаңа құрылғылар мен технологиялардың пайда болуына әкелді. Олар біздің өмірімізге барынша берік еніп, үйреншікті жағдайға айналады. Бүгінгі таңда мұндай гаджеттер арасындағы жетекші платформа-Android ОЖ.

Android көптеген құрылғыларда қолданылады. Бұл Смартфондар, планшеттер, Теледидарлар, смарт-сағаттар және басқа да гаджеттер. Әр түрлі есептеулер бойынша, 2020 жылы бұл операциялық жүйені смартфон иелерінің шамамен 85% пайдаланады, ал Android ОЖ-де смартфондарды пайдаланушылардың жалпы саны бүкіл әлем бойынша 2,5 миллиардтан астам адамға бағаланады.[10]

Android ОЖ-ны әзірлеуші Энди Рубин (Andy Rubin) ұялы телефондарға арналған Операциялық жүйе ретінде Android Inc атында дамыды. Бірақ 2005 жылы Google Android Inc-ті сатып алады. ол операциялық жүйені жаңа күшпен дамыта бастайды. Android үнемі дамып келеді және операциялық жүйемен бірге өңдеу құралдары дамиды.[12]

Қазіргі уақытта Java тілі ең көп таралған және танымал бағдарламалау тілдерінің бірі болып табылады. Тілдің алғашқы нұсқасы 1996 жылы Sun Microsystems компаниясының ішегінде пайда болды, кейіннен Oracle сіңірді. Java әртүрлі тапсырмалар үшін қолдануға болатын әмбебап бағдарламалау тілі ретінде ойластырылған. Қазіргі уақытта Java тілі үлкен жолды жүріп өтті, көптеген түрлі нұсқалар жарық көрді. Қазіргі нұсқасы-Java 18, ол 2022 жылдың 22 наурызында шыққан. Java қарапайым әмбебап тілден бүкіл платформаға және экожүйеге айналды, ол жұмыс үстелі қосымшаларын құрудан бастап, ірі веб-порталдар мен қызметтерді жазуға дейінгі бірқатар міндеттер үшін қолданылатын әртүрлі технологияларды біріктіреді. Сонымен қатар, Java тілі Көптеген құрылғыларға арналған бағдарламалық жасақтаманы жасау үшін белсенді қолданылады: қарапайым компьютерлер, планшеттер, смартфондар мен ұялы телефондар, тіпті тұрмыстық техника. Android мобильді ОЖ-нің танымалдылығын еске түсіру жеткілікті, оның көптеген бағдарламалары Java-да жазылған.[12]

Java тілінің басты ерекшелігі - оның коды алдымен платформадан тәуелсіз арнайы байт кодына аударылады. Содан кейін бұл байт кодын JVM виртуалды машинасы (Java Virtual Machine) орындайды. Осыған байланысты Java PHP немесе Perl сияқты стандартты интерпретацияланған тілдерден ерекшеленеді, олардың кодын аудармашы бірден орындайды. Сонымен қатар, Java-бұл C немесе C++сияқты таза құрастырылған тіл емес.[10]

Мұндай архитектура Java-да платформалық және аппараттық бағдарламалардың тасымалдануын қамтамасыз етеді, сондықтан мұндай

бағдарламаларды қайта құрастырусыз әртүрлі платформаларда - Windows, Linux, Mac OS және т.б. орындауға болады.

Java – C синтаксисі бар тіл және осыған байланысты C/C++ және C#тілдеріне жақын. Сондықтан, егер сіз осы тілдердің бірімен таныс болсаңыз, Java-ны меңгеру оңайырақ болады.[10]

Java-ның тағы бір басты ерекшелігі-ол қоқысты автоматты түрде жинауды қолдайды. Бұл дегеніміз, жадты бұрын қолданылған объектілерден қолмен босатудың қажеті жоқ, өйткені қоқыс жинаушы сіз үшін автоматты түрде жасайды.[7]

Java-объектіге бағытталған тіл. Ол полиморфизмді, мұрагерлікті, статикалық теруді қолдайды. Нысанға бағытталған тәсіл үлкен, бірақ сонымен бірге икемді, кеңейтілетін және кеңейтілетін қосымшаларды құру мәселелерін шешуге мүмкіндік береді.

Java бағдарламалау тілінде дамыту үшін бізге JDK немесе Java Development Kit деп аталатын арнайы құралдар жиынтығы қажет. Алайда, JDK-тің әртүрлі қосымшалары бар екенін атап өткен жөн, бірақ олардың барлығы бірдей тілді - Java-ны қолданады. Ең танымал екі іске асыру - Oracle JDK және OpenJDK. Олардың айырмашылығы неде?

Oracle JDK толығымен Oracle компаниясымен дамиды. OpenJDK-ті Oracle компаниясы да, бірқатар компаниялар бірлесіп дамытады.

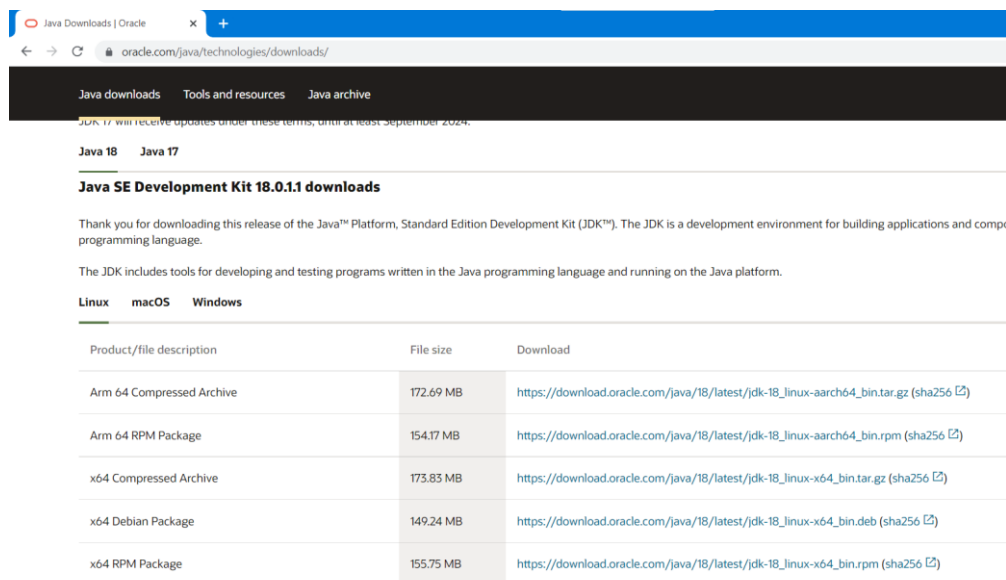
Лицензиялау тұрғысынан ең үлкен айырмашылықтар. Oracle лицензиясына сәйкес JDK-ны жеке қажеттіліктер үшін, сондай-ақ қосымшаларды әзірлеу, тестілеу және көрсету үшін ақысыз пайдалануға болады. Басқа жағдайларда (мысалы, қолдау алу үшін) жазылым түрінде коммерциялық лицензия қажет. Ал OpenJDK толығымен тегін.

Функционалдылық тұрғысынан Oracle JDK және OpenJDK мүмкіндіктерінің жиынтығы іс жүзінде ерекшеленбеуі керек. Бірақ өнімділік жоспары Oracle JDK OpenJDK-қа қарағанда біршама жылдам жұмыс істейтінін атап өтті. Сонымен қатар, кейбір әзірлеушілер OpenJDK сәл қате, ал Oracle JDK тұрақты екенін айтады. Осы жұмыста біз Oracle JDK қолданамыз.[10][7]

Сонымен, Java бағдарламаларын жасау үшін бізге JDK (Java Development Kit) әзірлеу үшін арнайы жинақ қажет. JDK құрамына Java бағдарламаларын құрастыруға, іске қосуға және басқа да бірқатар функцияларды орындауға мүмкіндік беретін бірқатар бағдарламалар мен утилиталар кіреді.[10][7]

JDK-тің тиісті нұсқасын Oracle ресми сайтынан жүктеп, орнатуға болады: <https://www.oracle.com/java/technologies/javase-downloads.html>

Бұл бетте біз операциялық жүйенің (Windows, MacOS немесе Linux) дистрибуциясын табамыз және жүктейміз (10-сурет):



Сурет 10. Oracle JDK орнату

Көптеген ОС үшін жүктеудің екі нұсқасы бар: орнатушы түрінде немесе орнатудың қажеті жоқ Мұрағат түрінде. Мысалы, менің ОС - Windows, сондықтан мен "ARM 64 RMP Package" тармағын таңдап, орнату бағдарламасын ұсынатын файлды жүктеймін.[10]

Android қосымшасында визуалды интерфейсті құрудың негізгі компоненті-бұл әрекет (әрекет). Көбінесе activity қосымшаның жеке экранымен немесе терезесімен байланысты, ал терезелер арасында ауысу бір әрекеттен екіншісіне ауысу ретінде жүреді. Бағдарламада бір немесе бірнеше әрекет болуы мүмкін. Мысалы, бос Activity көмегімен жобаны құру кезінде әдепкі жобаға бір Activity - MainActivity класы қосылады, ол бағдарлама жұмыс істей бастайды.[10][7]

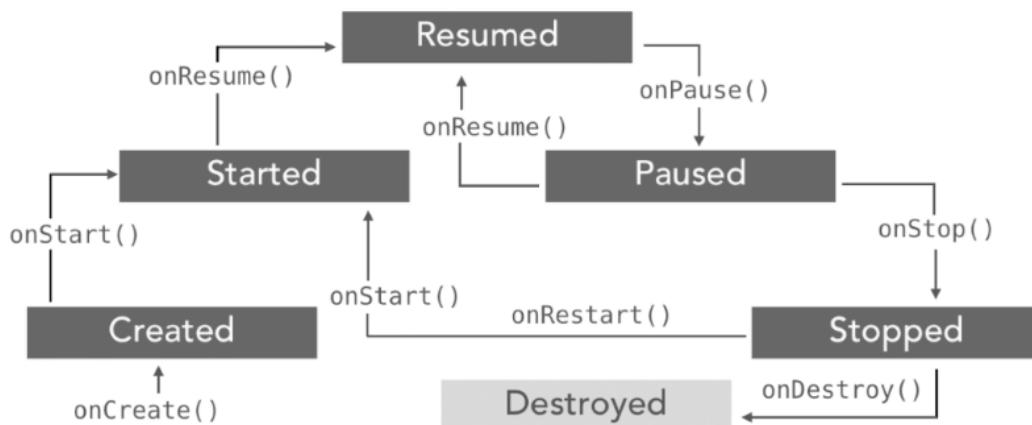
Барлық Android қосымшаларында жүйеде қатаң белгіленген өмірлік цикл бар. Пайдаланушы қосымшаны іске қосқан кезде жүйе бұл бағдарламаға жоғары басымдық береді. Әрбір қосымша жеке процесс ретінде іске қосылады, бұл жүйеге басқаларға қарағанда бір процестерге жоғары басымдық беруге мүмкіндік береді. Осының арқасында, мысалы, бір Android қосымшаларымен жұмыс жасағанда, кіріс қоңырауларды бұғаттамауға мүмкіндік береді. Қосымшамен жұмысты тоқтатқаннан кейін, жүйе барлық байланысты ресурстарды босатады және қосымшаны төмен басымдыққа айналдырады және оны жабады.[10]

Бағдарламадағы барлық әрекет нысандары жүйе back stack деп аталатын әрекет стегі ретінде басқарылады. Жаңа activity іске қосылған кезде ол стектің үстіне қойылып, жаңа activity пайда болғанша құрылғының экранына шығады. Ағымдағы әрекет өз жұмысын аяқтаған кезде (мысалы, пайдаланушы бағдарламадан кетеді), ол дестеден алынып тасталады және бұрын стекте екінші болған әрекет жұмысын жалғастырады.[10]

Іске қосылғаннан кейін activity жүйемен өңделетін және бірнеше қоңыраулар бар бірнеше оқиғалардан өтеді:

- protected void onCreate(Bundle savedInstanceState);
- protected void onStart();
- protected void onResume();
- protected void onPause();
- protected void onStop();
- protected void onDestroy();

11-суретте Activity компоненттерінің бір бірімен байланысын көре аламыз.



Сурет 11. Activity компонентінің жұмыс істеу қағидасы

Ең алдымен Android Studio өңдеу ортасында бағдарламамызды жасау үшін арнайы бастапқы (басты) бетін activity компонентін таңдаймыз.[10][7]

Проект бойынша келесідей экрандар тізімі қажет:

- Басты бірінші беті;
- Қолданушы нұсқасын таңдау беті;
- Авторизация;
- Басты қолдануға бейімделген экран (жолаушыларға және жүргізушілерге).

Басты қолданысқа иетін activity жолаушыларға тапсырыс жасауға мүмкіндік беретіндіктен және жолаушылардың негізгі жұмыс істеу тақтасы болғандықтан арнайы батырмалар тізімін ендіреміз.[10]

Жолаушылардың батырмалары:

- Тапсырыс беру батырмасы;
- Жөндеу батырмасы(настройки);
- Шығу.

Жүргізушілердің батырмалары:

- Тапсырыс қабылдау;
- Жөндеу батырмасы(настройки);
- Шығу.

Қосымша жолаушыларға ұсынылған батырмалар «ЖОЛАУШЫҒА ҚОҢЫРАУ ШАЛУ» және жүргізушілерге қоңырауды қабылдау мүмкіндігі керек.[10]

Проектінің мәліметтер қорымен жұмыс істеу үшін арнайы Google ұсынатын Firebase технологиясын қолданамыз.

Firebase мәтіндік деректерді JSON форматында сақтайды және деректерді оқу, жаңарту және алудың ыңғайлы әдістерін ұсынады. Сондай-ақ, Firebase пайдаланушыларды тіркеуге және авторизациялауға, сессияларды (уәкілетті пайдаланушылар) сақтауға, cloud Storage арқасында оңай қол жеткізуге мүмкіндік беретін медиа файлдарға көмектеседі.[11]

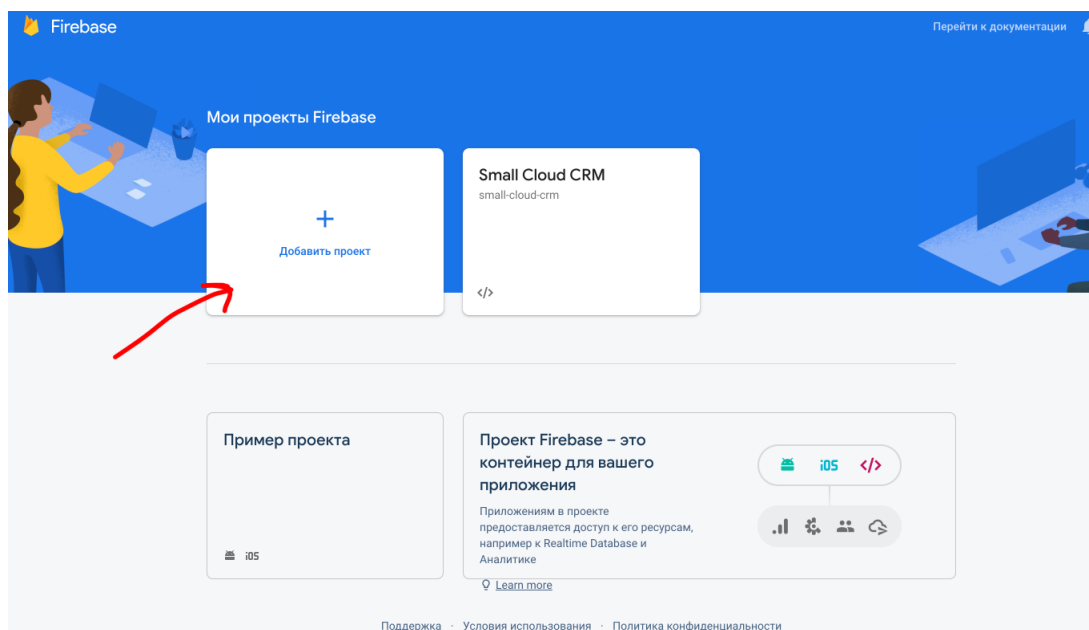
Python және JavaScript бағдарламалау тілдері икемді (және өте танымал). Осының арқасында біз (әзірлеушілер) қысқа мерзімде веб-сайтты, веб-қосымшаны, мобильді немесе тіпті жұмыс үстелі қосымшасын құрудың қуатты құралын аламыз.

Firebase сізге осы жылдамдықты сақтауға мүмкіндік береді. Басқа нәрселерге алаңдамаудың қажеті жоқ (мәліметтер базасын құру, қабылдау және қабылдау API жазу). Сервердің бүкіл бөлігі осы қызметтің иығына түседі. Осындай қуатты құралдың арқасында сіз кез-келген модульдің функционалдығын бір кеште оқып, деректерді сақтай отырып, тапсырыс берушіге функционалды мысал көрсете аласыз.

Егер сізде кішкентай SPA қосымшасы немесе қарапайым мобильді қосымшасы болса, онда ол кейбір мәтіндік мәндерді басқаруға азаяды, Сіз Firebase-ді негізгі мәліметтер базасы ретінде пайдалана аласыз. Тіпті төлеудің қажеті жоқ. [11]

Firebase Басқару тақтасы

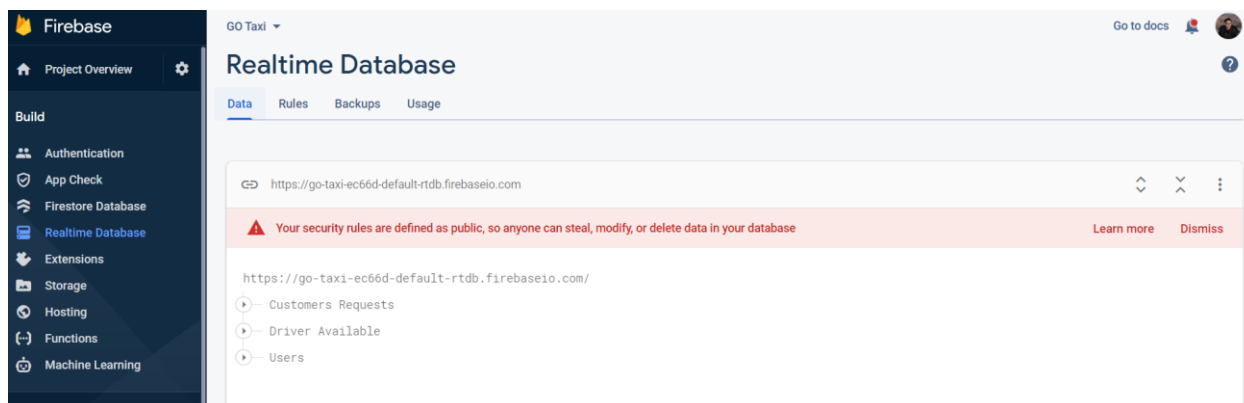
Алдымен сіз сайтқа өтіп, Firebase-ке тіркелуіңіз керек. Жылдам тіркелу үшін Google поштаңызды пайдалануға болады (12-сурет).[11]



Сурет 12. Firebase технологиясы

"Жобаны қосу" үлкен түймесін басыңыз. Қажет болса, Google аналитикасын қосыңыз. Сізден жобаның орналасқан жерін таңдау сұралуы мүмкін, тек өз еліңізді таңдаңыз.

Консоль ішінде бізді тек 3 нәрсе қызықтырады: жоба параметрлері, мәліметтер базасы және аутентификация. 13-суретте «Realtime Database» мәліметтер қорын көрсек болады. [11]



Сурет 13. Realtime Database

3.2 Мобильді қосымшаның функционалы

14-суретте бейнеленген бастапқы бетте бізде қосымшаның логотипі бренд аты activity бейнесі көрінеді. Бұл экран бір секундтан аспай көрсетіледі. Ол логотипті, компанияның немесе өнімнің брендін ұсынады. Біздің жағдайда қосымшаның логотипі мен атауы.



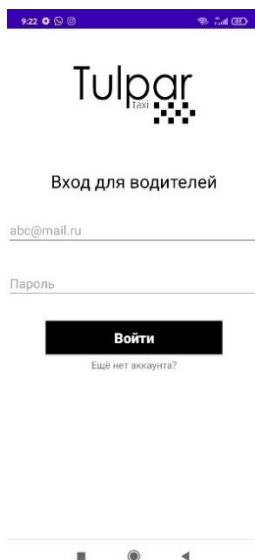
Сурет 14. Қосымшаның бірінші беті

Келесі экран пайдаланушы түрін таңдауға мүмкіндік береді. Бұл жүргізуші мен клиент (15-сурет).

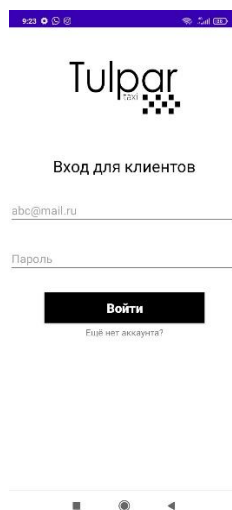


Сурет 15. Пайдаланушы типін таңдау

Келесі экран – авторизация экраны. Бұл экран пайдаланушы өзінің авторизациялық деректерімен кірмеген жағдайда көрсетіледі. Мұнда пайдаланушы өз атымен кіре алады немесе тіркеле алады. Авторизация экраны пайдаланушылардың екі түрі үшін де бірдей (16-сурет), (17-сурет).



Сурет 16. Жүргізушілердің авторизациялау беті



Сурет 17. Жолаушылардың авторизациялау беті

Келесі беттерде клиенттерге және жүргізушілерге арналған экран. Бұл Экранда Google Map API арқылы алынған картасы. Қолданушылар GPS көмегімен өзінің орналасқан нүктесін көре алады. Сол арқылы біздің деректер қорымыздағы ақпаратты өзгерте аламыз. Ол клиенттің немесе жүргізушінің мекен жайы.

Алынған ақпарат бойынша жолаушы мен жүргізуші арасындағы қашықтықты есептеу мүмкіндігі бар. Солайша жүйе жолаушылар мен жүргізушілер арасындағы қашықтықты қолданушыларға көрсетеді.

Және жоғарыдағы панельде “Настройки” және “Выйти” батырмаларын байқасақ болады (18-сурет).

«Выйти» - батырмасы қолданушылардың жүйеден шығуға мүмкіндік береді. Жүйеден шыққан қолданушылар фондық режимде ақпараттары алынбайды (18-сурет).



Сурет 18. Жүргізушілердің басты беті

Клиенттердің жүргізуші шақыру экранында “Вызвать такси” батырмасы орнатылған. Батырма іске қосылғаннан кейін жақын арадағы орналасқан жүргізушілердің біреуіне тапсырыс түседі. Тапсырыс қабылдаған жүргізушіге клиенттің ақпараты түседі. Ол орналасқан мекен жайы және телефон нөмірі (19-сурет).



Сурет 19. Жолаушылардың басты беті

ҚОРЫТЫНДЫ

Дипломдық жұмысқа «такси компаниясына ақпараттық жүйені өңдеу» байланысты, пәндік аймаққа қатысты, анализ жүргізілді. Анализ нәтижесінде берілген ақпаратты синтездей отырып ақпараттық жүйеге негізгі талаптары белгіленді.

UML диаграммалары арқылы проекттің негізгі концепциясы мен структурасы көрсетілген. Концепцияға сай болатын қажетті технологиялар зерттелген. Қолданылған технологиялар нәтижесінде ақпараттық жүйе өңделді.

Қазіргі заманда өзекті құрал мобильді қосымша болып саналатындықтан, қарапайым пайдаланушыларға арналған мобильді қосымша құрылды.

Қолданылған мәліметтер мен технологиялардың нәтижесінде, өңделген ақпараттық жүйе, кәсіпті дамуға үлесін қосады деп ойлаймын.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Как предлагают регулировать рынок такси. [Электрондық ресурс].
Сілтеме: <https://nalogikz.kz/docs/kak-predlagayut-regulirovat-rynok-ta.html>
2. Анализ рынка услуг такси в Казахстане в 2016-2020 гг., прогноз на 2021-2025 гг. (демо), (<https://marketing.rbc.ru/research/42668/>) 8-13 бет.
3. Динамикалық баға беру және тапсырыстар тарату. *Dynamic Pricing and Matching in Ride-Hailing Platforms* — N.Korolko, D.Woodard, C.Yan, H.Zhu (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3258234) 13.11.2019
4. Принцип работы Яндекс-такси, Uber и Gett. [Электрондық ресурс].
Сілтеме: <https://vist-m.ru/articles/concept>.
5. Официальный сайт Yandex Go. [Электрондық ресурс]. Сілтеме: https://taxi.yandex.kz/ru_kz/almaty/tariff
6. Официальный сайт InDriver. [Электрондық ресурс]. Сілтеме: <https://indriver.com/ru/city>
7. Д.Гриффитс, Д.Гриффитс. Head First. Программирование для Android. – Питер: 2018.
8. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. [Электрондық ресурс].
Сілтеме: habr.com/ru/post/566218/.
9. ER диаграммы. [Электрондық ресурс]. Сілтеме: <https://www.lucidchart.com/pages/ru/erd>.
10. Программирование под Андроид на Java. [Электрондық ресурс].
Сілтеме: <https://metanit.com/java/android>
11. Официальный сайт Firebase. [Электрондық ресурс]. Сілтеме: <https://firebase.google.com>.
12. Д.Блох, Java: эффективное программирование, 3-е издание. Диалектика, Мәскеу-2019.

А қосымшасы

Customer Login

```
package com.example.mytaxi;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class CustomerRegLoginActivity extends AppCompatActivity {

    TextView customerStatus, question;
    Button signInBtn, signUpBtn;
    EditText emailET, passwordET;

    FirebaseAuth mAuth;
    DatabaseReference CustomerDatabaseRef;
    String OnlineCustomerID;

    ProgressDialog loadingBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_customer_reg_login);
    }
}
```

А қосымшаның жалғасы

```
customerStatus = (TextView)findViewById(R.id.statusCustomer);
question = (TextView)findViewById(R.id.accountCreateCustomer);
signInBtn = (Button) findViewById(R.id.signInCustomer);
signUpBtn = (Button) findViewById(R.id.signUpCustomer);
emailET = (EditText) findViewById(R.id.customerEmail);
passwordET = (EditText) findViewById(R.id.customerPassword);

 mAuth = FirebaseAuth.getInstance();

loadingBar = new ProgressDialog(this);

signUpBtn.setVisibility(View.INVISIBLE);
signUpBtn.setEnabled(false);

question.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        signInBtn.setVisibility(View.INVISIBLE);
        question.setVisibility(View.INVISIBLE);
        signUpBtn.setVisibility(View.VISIBLE);
        signUpBtn.setEnabled(true);
        customerStatus.setText("Регистрация для клиентов");
    }
});

signUpBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String email = emailET.getText().toString();
        String password = passwordET.getText().toString();

        RegisterCustomer(email, password);
    }
});

signInBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
String email = emailET.getText().toString();
```

А қосымшаның жалғасы

```
String password = passwordET.getText().toString();

SignInCustomer(email, password);
}
});
}

private void SignInCustomer(String email, String password) {

    loadingBar.setTitle("Вход для клиентов");
    loadingBar.setMessage("Пожалуйста, дождитесь загрузки");
    loadingBar.show();

    mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful())
        {
            Toast.makeText(CustomerRegLoginActivity.this, "Успешный вход.",
Toast.LENGTH_SHORT).show();
            loadingBar.dismiss();

            Intent customerIntent = new Intent(CustomerRegLoginActivity.this,
CustomersMapActivity.class);
            startActivity(customerIntent);
        }
        else
        {
            Toast.makeText(CustomerRegLoginActivity.this, "Произошла
ошибка, попробуйте снова", Toast.LENGTH_SHORT).show();
            loadingBar.dismiss();
        }
    }
});
}

private void RegisterCustomer(String email, String password)
{ loadingBar.setTitle("Регистрация клиента");
```

А қосымшаның жалғасы

```
loadingBar.setMessage("Пожалуйста, дождитесь загрузки");
loadingBar.show();

mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful())
        {
            OnlineCustomerID = mAuth.getCurrentUser().getUid();
            CustomerDatabaseRef =
            FirebaseDatabase.getInstance().getReference()
                .child("Users").child("Customers").child(OnlineCustomerID);
            CustomerDatabaseRef.setValue(true);

            Intent customerIntent = new Intent(CustomerRegLoginActivity.this,
            CustomersMapActivity.class);
            startActivity(customerIntent);

            Toast.makeText(CustomerRegLoginActivity.this, "Регистрация
прошла успешно", Toast.LENGTH_SHORT).show();
            loadingBar.dismiss();
        }
        else
        {
            Toast.makeText(CustomerRegLoginActivity.this, "Ошибка",
            Toast.LENGTH_SHORT).show();
            loadingBar.dismiss();
        }
    }
});
}
```

Б қосымшасы

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/welcome"
    tools:context=".CustomerRegLoginActivity">
```

```
<EditText
    android:id="@+id/customerEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/statusCustomer"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:hint="@string/email"
    android:inputType="textEmailAddress" />
```

```
<EditText
    android:id="@+id/customerPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/customerEmail"
    android:hint="@string/pass"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:inputType="textPassword" />
```

```
<Button
    android:id="@+id/signInCustomer"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/customerPassword"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"
    android:background="#000000"
    android:text="@string/signIn"
```

Б қосымшаның жалғасы

```
android:textAllCaps="false"
```

```
android:textColor="@android:color/white"  
    android:textSize="20sp"  
    android:textStyle="bold" />
```

```
<Button  
    android:id="@+id/signUpCustomer"  
    android:layout_width="250dp"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/signInCustomer"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="30dp"  
    android:background="#FFFFFF"  
    android:text="@string/signUp"  
    android:textAllCaps="false"  
    android:textColor="@android:color/black"  
    android:textSize="20sp"  
    android:textStyle="bold"/>
```

```
<TextView  
    android:id="@+id/statusCustomer"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="200dp"  
    android:text="Вход для клиентов"  
    android:textColor="@android:color/black"  
    android:textSize="25sp"  
    android:textStyle="normal" />
```

```
<TextView  
    android:id="@+id/accountCreateCustomer"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_below="@+id/signInCustomer"  
    android:layout_marginTop="5dp"  
    android:text="@string/createQuestion" /></RelativeLayout>
```


**Университеттің жүйе администраторы мен Академиялық мәселелер
департаменті директорының ұқсастық есебіне талдау хаттамасы**

Академиялық жетекшінің ұқсастық туралы есебі

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагиаттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Авторы: Сағатов Біржан Маратұлы

Тақырыбы: Транспорттық компанияға (такси) арналған ақпараттық жүйесін өндеу

Жетекшісі: Оған Аткалді

1-ұқсастық коэффициенті (30): 6,41%

2-ұқсастық коэффициенті (5): 3,34%

Әріпті ауыстыру: 1

Интервалдар: 0

Микрокеңістіктер: 0

Ақ белгілер: 117

Есепті талдағаннан кейін:

- ☐ Жұмыста анықталған ұқсас алынған сөздерде плагиат белгілері жоқ. Осыған байланысты бұл жұмысты өз бетінше орындаған және оны қорғауға жіберемін.
- ☐ Жұмыста анықталған ұқсас сөздерде плагиат белгілері жоқ, бірақ олардың ұқсастығының шамадан тыс саны жұмыстың құндылығына және оның авторының дербестігінің болмауына қатысты күмән тудырады. Осыған байланысты жұмыс қайта редакциялансын.
- ☐ Жұмыста көшіріп алу және плагиат белгілері бар жұмысты жасыру әрекеттерін көрсете отырып, плагиаттық белгілерге немесе мәтінді қасақана бұрмалауға әкелетін әрекеттер бар. Осыған байланысты, жұмыс қорғауға жіберілмейді.

Негіздемесі:

Күні:

техн.ғыл.маг., лектор



Оған Аткалді

**Университеттің жүйе администраторы мен Академиялық мәселелер
департаменті директорының ұқсастық есебіне талдау хаттамасы**

**Бөлім меңгерушісінің/ құрылымдық бөлімшесінің басшысының ұқсастығы
туралы есебі**

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагиаттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Авторы: Сағатов Біржан Маратұлы

Тақырыбы: Транспорттық компанияға (такси) арналған ақпараттық жүйесін өңдеу

Жетекшісі: Оған Аتكелді

1-ұқсастық коэффициенті (30): 6,41%

2-ұқсастық коэффициенті (5): 3,34%

Әріпті ауыстыру: 1

Интервалдар: 0

Микрокеңістіктер: 0

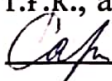
Ақ белгілер: 117

Есепті талдағаннан кейін:

- ☐ Жұмыста анықталған ұқсас алынған сөздерде плагиат белгілері жоқ. Осыған байланысты бұл жұмысты өз бетінше орындаған және оны қорғауға жіберемін.
- ☐ Жұмыста анықталған ұқсас сөздерде плагиат белгілері жоқ, бірақ олардың ұқсастығының шамадан тыс саны жұмыстың құндылығына және оның авторының дербестігінің болмауына қатысты күмән тудырады. Осыған байланысты жұмыс қайта редакциялансын.
- ☐ Жұмыста көшіріп алу және плагиат белгілері бар жұмысты жасыру әрекеттерін көрсете отырып, плагиаттық белгілерге немесе мәтінді қасақана бұрмалауға әкелетін әрекеттер бар. Осыған байланысты, жұмыс қорғауға жіберілмейді.

Негіздемесі:

Күні:

КАӨЖС кафедрасы меңгерушісі
т.ғ.к., ассоц. профессор
 Р.Ж.Сатыбалдиева

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Дипломдық жұмысқа (жобаға)

ПІКІР

Т. А. Ә.: Сағатов Біржан Маратұлы.

Мамандық: 5В070300 – «Ақпараттық жүйелер»

Дипломдық жұмыс тақырыбы: «Транспорттық компанияға (такси) арналған ақпараттық жүйесі өңдеу»

Сағатов Біжанның дипломдық жұмысы таксикомпаниясына арналған ақпараттық жүйесі.

«Tulpar Taxi» мобильді қосымшасында ағымдағы такси шақыру мүмкіндігі ба, сонымен қатар басқа агрегаторлардан такси шақыру мүмкіндігі бар.

Сағатов Б. дипломдық жұмысында жоба алды зерттеулерінде жұмыс ерекшелігін көрсете білген, зерттеу тақырыбының өзектілігін нанымды және дәлелді түрде көрсеткен.

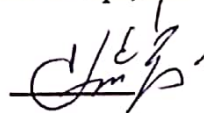
Дипломдық жобаны аяқтау барысында Сағатов Б. барлық тағайындалған тапсырмаларды орныдады: такси нарығындағы агрегаторларды орталандыру мүмкіндігінің жобасын көрсету, UML диаграммасы арқылы қосымшаның жүйесін бейнелеу.

Жалпы сипаттай отырып, студенттің дипломдық жұмысындағы зерттеу логикасы, тараулар мен бөлімдердің бірізділігі мен мазмұны тақырыпты терең әрі сапалы ашуға мүмкіндік берді.

Ғылыми жетікші:

тех.ғыл.магистрі, лектор

Оған А.



«___» _____ 2022 ж.

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу
университетінің
5B070300 – «Ақпараттық жүйелер» мамандығының 4 курс студенті
Сағатов Біржан Маратұлының
«Шағын бизнесте ақпараттық жүйесінің компонентін жасау»
тақырыбына жазылған дипломдық жұмысына

СЫН ПІКІР

Дипломдық жұмыстың агрегаторлардың орталаңдырған жүйесін өңдеу такси компанияларына арналған автоматтандырылған құралды жасау қарастырылған.

Дипломдық жұмыста такси компанияларының Қазақстан нарығының мәліметтерінен негізделген. Такси нарығының статистикалық мәліметтері ұсынылған. Нарықтағы InDriver, Uber, Yandex Taxi агрегаторлары зерттелген. Жұмыста агрегаторлардың жұмыс істеу қағидалары және бағаларын қарастырдық. Сервистің өзектілігі және ерекшеліктерін көрсетілген.

UML диаграммалары арқылы дипломдық проекттің ұйымдық құрылымы графикалық түрде көрсетілген.

Жобаны өңдеуге арналған бағдармалау тілдері мен технологиялардың артықшылықтары жазылған. Жобаның қолдану процесстері көрсетілген.

Ескертпелер:

Агрегаторлардың API ақылы болғандықтан проект толығымен өңделмеген. Проекттің ауқымдылығы жоғары.

Диплом жұмыс квалификациясы жұмыстарға қойылатын талаптарға сәйкес келеді және қорғауға ұсынылады, ал авторы, студент Сағатов Біржан 5B070300 – «Ақпараттық жүйелер» мамандығы бойынша «бакалавр» академиялық дәрежесін беруге болады деп есептеймін. Дипломдық жоба "92%" бағасына лайық.

Рецензент:

Ғұмарбек Дәукеев атындағы
Алматы энергетика және
байланыс университеті,
«Ақпараттық жүйелері және
киберқауіпсіздік» кафедрасының
доценті, техника ғылымдарының
кандидаты



Тусупова Б.Б.

Қолтаңбаны растаймын		
Подпись заверяю		
Қызметі	аты-жөні	
«23»	05	2022ж.