


Отчет подобию



Университет:	Baibazov University
Название:	Технология Final Exam
Автор:	Тайболат Нуртұрман
Кандидаттар:	Аман Адилябекова
Дата отчета:	2012-05-08 10:33:31
Коэффициент подобию № 1: <input type="checkbox"/>	0,1%
Коэффициент подобию № 2: <input type="checkbox"/>	0,0%
Длина фразы для коэффициента подобию № 2: <input type="checkbox"/>	25
Количество слов:	4 653
Число вхождений:	37 343
Адреса пропущенных при проверке:	
Количество обнаруженных пропусков: <input type="checkbox"/>	77

Ғылыми жетекші  А. К. Адильбекова

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Мамандығы 5B060200 – Есептеу техникасы және бағдарламалық
қамтамасыз ету

Студент Тасболат Нұрсұлтан

Тақырыбы: «Android платформасында "Qosh keldiniz KZ" Қазақстанның
туристік аймақтарына мобильді қосымша құру»

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ШІКІРІ

Дипломдық жобада Қазақстанның туристік жерлеріне арнайы арналып
заманауи талаптарға сай, көп тілде қосымша жасалып отыр.

Дипломдық жоба Android платформасында кез-келген құрылғыда
жұмыс атқара отырып, шет елден келген және елдегі туристтер үшін
Қазақстандағы туристік аймақтарына арналған ақпараттық бағдарлама.

Дипломдық жұмыс кіріспеден, үш бөлімнен, қолданылған әдебиеттер
тізімінен, қорытындыдан тұрады. Бірінші бөлімде мәселенің қазіргі
замандағы жағдайына шолу және оны талдау жасалынады. Екінші бөлімде
программалау тілдерін пайдалану ерекшеліктері жобаланған. Үшінші
бөлімде туристтер үшін жасалған қосымша жайлы толық ақпарат қамтылып
отыр.

Дипломдық жобаның Қазақстанның туризм саласына қосатын үлесі
көп. Жұмыс материалы жоспарлы түрде, қойылған талаптарға сай, тақырып
төңірегінде жазылған. Студенттің жұмысты рәсімдеудегі біліктілігін,
талдаушылық қабілетін көруге болады. Осы жұмыста негізгі өзекті
мәселелерді шеше отырып, зерттеу жұмысында оңтайлы нәтижелер алып,
жұмысты қойылған талаптарға сай орындап шықты.

Қорытындылай келгенде, осы зерттеу жұмысын орындау барысында
Тасболат Нұрсұлтан Алматылы жетекшінің көмегімен жұмысын
ойындағыдай атқарып шықты.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес
деп есептей отырып Тасболат Нұрсұлтан Алматылы «Android
платформасында "Qosh keldiniz KZ" Қазақстанның туристік аймақтарына
мобильді қосымша құру» атты тақырыптағы дипломдық жұмысын
жауапкершілікпен орындады, 5B060200– «Есептеу техникасы және
бағдарламалық қамтамасыз ету» мамандығы бойынша «Жаратылыстану
бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ

«Программалық инженерия»
кафедрасының лекторы «30»

 А.К. Адильбекова

04 2019 жыл

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Тасболат Нұрсұлтан Алматыұлы

Android платформасында "Qosh keldiniz KZ" Қазақстанның туристік аймақтарына мобильді қосымша құру

Дипломдық жұбаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,

ассистент-профессор

 Р. Юнусов

" 30 " 04 2019 ж.

Дипломдық жұбаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: Android платформасында "Qosh keldiniz KZ" Қазақстанның туристік аймақтарына мобильді қосымша құру

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

Орындаған

Тасболат Н. А.

Ғылыми жетекші

техн. ғыл. магистрі, лектор

 А. Адильбекова

" 30 " 04 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӨТБАЕВ УНИВЕРСИТЕТІ

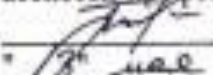
Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

SB070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

БЕКІТЕМІН

Кафедра меңгерушісі
тех. ғыл. кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
"13" мае 2019 ж.

**Дипломдық жоба орындауға
ТАПСЫРМА**

Білім алушы Тасболат Нұрсұлтан Алматылы

Тақырыбы: Android платформасында "Qash keldiniz KZ" Қазақстанның туристік аймақтарына мобильді қосымша құру

Университет ректоры бұйрығының № 1841-б "14" наурыз 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі «13» «05» 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Ұсынылатын дипломдық жобаны құру барысында шет елден келген туристтер үшін жасалған мобильді қосымша

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) жобаның талдауы;

б) жобалау бөлімі;

в) жасау құралдары;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма

е) Б Қосымшасы – бағдарлама мәтіні.

Жобаның презентациялық 15 слайды ұсынылған.

Ұсынылған негізгі әдебиеттер: 16 атау.

Дипломдық жобаны орындау
КЕСТЕСІ

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

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

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

Ғылыми жетекші _____ *А.А. Таурбекова* Адильбекова А.

Тапсырманы орындауға қабылдап алған студент _____ *Н.А. Тасболат* Тасболат Н. А.

Күні

«3» 10 2018 ж.

АҢДАТПА

Дипломдық жоба Android платформасында "Qosh keldiniz KZ" Қазақстанның туристік аймақтарына мобильді қосымша құру жұмыс барысында көптеген зерттеулер жүргізілді. Дипломдық жобаның басты мақсаты – "Qosh keldiniz KZ" мобильді қосымшасы, ол көптен күткен мобильді қосымша, осы қосымша көмегімен атақты туристік жерлермен және Қазақстанның туристік аймақтары туралы қызықты ақпараттармен таныс боласыздар. "Qosh keldiniz KZ" жобасы пайдаланушыларға саяхаттау кезін жеңілдетуге көмектеседі. Дипломдық жобаны орындау кезінде PHP және Java тілі және мобильді құрылғылардың Android Studio құралы қолданылды.

АННОТАЦИЯ

Дипломный проект на платформе Android "Qosh keldiniz KZ" в ходе работы по созданию мобильного приложения в туристические зоны Казахстана было проведено множество исследований. Основная цель дипломного проекта- Создание мобильного приложения "Qosh keldiniz KZ", которое уже давно ожидалось. Мобильное приложение, с помощью которого вы сможете ознакомиться с известными туристскими местами и интересными туристскими регионами Казахстана. При выполнении дипломного проекта использовался язык PHP и Java и инструмент Android Studio мобильных устройств.

ANNOTATION

Diploma project on the Android platform "Qosh keldiniz KZ" during the work on the creation of a mobile application in the tourist areas of Kazakhstan, a lot of research was conducted. The main goal of the diploma project is to Create a mobile application "Qosh keldiniz KZ", which has long been expected. Mobile application with which you can get acquainted with famous tourist places and interesting tourist regions of Kazakhstan. During the graduation project, the language PHP and Java and the tool Android Studio mobile devices were used.

МАЗМҰНЫ

Кіріспе	8
1 Негізгі бөлім	9
1.1 Жобаға шолу	9
1.2 Туристерге арналған қосымшалар	10
1.3 «Qosh keldniniz KZ» мобильді қосымшасын талдау	11
2 Жобалау бөлімі	14
2.1 UML модельдеу тілімен қосымшаны жобалау	14
2.2 Қосымшаның архитектурасы	20
3 Қосымшаны әзірлеудің бағдарламалық құралдары	24
3.1 Жобаны іске асыру ерекшеліктері	24
3.2 «Qosh keldniniz KZ» мобильді қосымшасының интерфэйсі	33
Қорытынды	36
Пайдаланылған әдебиеттер тізімі	37
А Қосымшасы. Техникалық тапсырма	38
Б Қосымшасы. Бағдарлама мәтіні	43

КІРІСПЕ

Дәл қазіргі сәт – адамзат дамуындағы ең қатты серпіліс алған заман деп айтуға тұрарлық кезең. Өйткені дамып жатқан бүкіл әлем бір-бірімен тікелей байланыста деп айтуға да болады. Бұл сөзге тағы бір дәлел болып табылатын жәйт, ол адамдадың көзбен көру арқылы өзінің қызуғушылықтарына қол жеткізу.

Басқа ел аралап, табиғатымен, салт-дәстүрімен танысып және осының бәрін өз көзімен көру дерлік, әр замандағы адамдардың қызығушылықтарының бірі.

Өткен замандағы саяхаттаушылар мен қазіргі замандағыларды салыстыратын болсақ, орасан зор айырмашылықтар бар. Қазіргі замандағы саяхаттаушылар кез-келген жерге баруға және көруге мүмкіншілік бар. Бірден-бір оған себеп дамыған заман ақпараттық технологиялар дамыған заманы.

Ақпараттық технологиялар заманы дегеніміз бір ауыз сөзбен ол – адамдардың өмірне технологияның араласуымен жеңілдетілген өмір. Бүгінгі таңда саяхаттаушыларға арналған көптеген қосымшалар мен веб-парақшалар бар. Және менің дипломдық тақырыбым Қазақстанға келген саяхатшыларға арналған турситтік мекендер үшін жасалған мобильді қосымша.

Бұл тақырыпты таңдауымның басты себебі ол – елімізге келген, көбінесе бірінші рет келіп отқан туристтерге көптеген ақпарат алу, тез арада керекті бағытты табу үшін жасалды.

Қазақстан жері кең-байтақ, өзен-көлді, жері естеліктерді әлі күнге дейін сақтап келе жатқан байырғы дамыған мемлекет деп айтуға болады. Сол себепті кәзіргі таңда әлемдегі туристтердің қызығушылығы Орталық Азия мен Азия мемлекеттерінде. Соның ішінде Қазақстанға да ағылып келіп жатқан туристтер үшін қазақ жеріндегі туристтік бағыттарды көріп, таңдап және Қазақстанға келген уақытында көптеген сурақтарға жауап алуға болатын қосымша Андроид ОЖ жасалды.

1 Негізгі бөлім

1.1 Жобаға шолу

Турситтерге арналған мобильді қосымшасының негізгі атқаратын қызметі болып көп пайдалылығы болу керектігі мәлім. Кез-келген елде, кез-кез келген туристтер көптеген қиыншылықтармен жолығады. Елге келген кезде ұшақтан түскен кездегі сәттен бастап.

Бүгінгі таңдағы туризм жағынан және елдің экономикасының мақсаты жағынан дамып оның шыңында жүрген мемлекеттер бар. Ал Қазақстан мемлекетіміз бұл саланың әлі дәмін енді татып жүр. Сол себепті қазір бұл яғни туризм саласының дамытудың маңызы зор.

Елге бірінші рет болсын алдын келеген саяхатшы өзінің сезінгенін өз еліне жайып салатыны белгілі. Ал енді осыны біздің елге келген саяхатшылардың ойынан шығу керек. Кәзіргі таңда көптеген жұмыстар атқарылып та жатыр.

Интерфэйстің түсінікті болуының маңызы басты шарт болып табылады, себебі қолданушылар үшін ынғайлылық пен көзге ынғайлылығы маңызды. Қолданушы ортаның кез-келген санаптағы адамдар қатары болу мүмкін, саяхаттау кезінде кез-келген сатыдағы адамдар болады.

Ғаламтор арқылы мәліметтер іздеп табу ешкімге қиындық соқтырмайтыны мәлім, сонда да болатын жәйіттер қосымшаның өзіндің болғаны жөн. Ақпаратты алу үшін, батырманы басу арқылы, қалаған мәліметтер алу маңызды. Түсінікті жасалған бағдарлама ғана сұранысқа ие болады. Сондықтан да жобаны бастау алған кезде барлық жәйіттерді қарастыру өте маңызды.

Жұмыстың жобалаудағы тағы да басқа көптеген жұмыстар бар. Қосымша масштабы соның ішінде болады. Пайдаланушы алғашқы рет қосымшаны қолданған кезде оның көзге түсетін қарапайымдалығы мен тез жұмыс жасауы. Ары қарай контексті масштабпен жұмыс атқаруына тоқталамыз.

Жоспарлау кезінде жобалау маңызды және осынын бірі болып табылатын жәйіт, қолданған кезде қосымшаны ішінде оңайлықпен кез-келген бөлігіне кіруі болады.

Жобалаудағы маңыздылықтың басқа да маңыздылығы сәнін қалыптастыра отырып, оның қолданыушының көңілінен шығу маңыздылығына ие болад. Айта кетсек оның ішіндегі сәнімен қатар, сыртқы сәне де маңызды орын алады. Санымен ғана көрсетпей сапа жағыда болуы басты орында.

Құрылғының маңыздылығы және жасалған қосымшаға атқарылатын міндеттері орасан зор. Осы жобалаудың барлық түйінін қарап өтуі маңыздылығы көп екенін барлығына мәлім. Жасалған артықшылықтарының барлық түрлі жағдайында болатын мәліметтер арқылы өтетін дерлік мәлімет. Осы арқылы

жасалған жобалау әдістерін дұрыс орындау және тура орындау маңыздылығы да барын ұмытпау керек.

Жобаны жобалау өзіндік стандарттарға және көптеген басқармалық жәйіттерге иеленеді.

Соңында бұл жобалаудың өзіндік нәтежиесін көрсетуге болады. Қосымшаның барлық дерлік жұмысы маңызды. Жобалау арқылы өтетін маңыздылықтардың барлығын көңілден шығатын, қолданықа ие қосымша жасалып шығады.

1.2 Туристтерге арналған қосымшалар

Ғаламтор жеткелінде Қазақстанға келген туристтерге арналған мәліметтер өте көп. Сонымен қатар ол веб парақшалық мәліметтер. Ал бұл жобаның айналасындағы, яғни Андроид ОЖ жасалған жобаларға тоқталатын болсақ оны Play Market-ке шолу жасауымыз қажет.

Мобильді қосымшаларды көру арқылы өзіме идеялар мен жетіспеушіліктерді көрдім. Жалпы алғанда көптеген қосымшалардың бағыты Алматы, Астана қалаларына арналған қосымшалар. Бұл бағдарлмалардың идеялары бір болғанмен, мақсаттары әр түрлі болып келеді. Және әр қосымшаның ұқсастығы да бар.

1.1-кесте – Qosh keldniz KZ қосымшасын салыстыру

	Qosh keldniniz KZ	Басқа қосымшалар
Жұмыс жасауы	Жақсы	Жақсы
Поддержка версий Android	+	+
Material Design	+	+
Виджеттер	-	-
Қосалқы үндемелер	+	+
Жөндеу	+	-
Көптілділік	+	-

Жасалған жұмыстардың бәрі өзіндік алатын орны ерекше, өйткені сол арқылы даму, алға қарай ұмтылыс болмақ. Келесі кесте арқылы басқа қосымшалардың артықшылықтаы мен кемшілігін салысытып 1.1-кестеде көреміз.

Кестеде көрсетілгендей басқа қосымшалардың мақсаты бір болғандықтан көп жерде ұқсастықтар табуға болады. Бірақтан өзіндік айтарлықтай ерекшеліктері бар.

1.3 «Qosh keldiniz KZ» мобильді қосымшасын талдау

Шет елден келген саяхатшылар үшін жасалған "Qosh keldiniz KZ" ақпараттық техникалық базада жасалған қосымша болып табылады. Қосымшаны жасаудағы басты мақсат болып саяхатшылар келген кезде елдің ішінде өзін жоғалтып алмай оңай жүріп-тұруы болып табылады. Және болған сұрақтар арқылы өзінің жолын тауып, қауыптер мен сұрақтардың жауап алу.

Қосымшаның бөліктерден бөліп көрсетуге болады. Бірінші бөлімінде басты бет, яғни демалыс түрлері. Екінші ретте қосымша мәзірдегі қосымшалар түрі. Үшінші-демалыс орнының сипаттамасы, суреттері.

Қосымша жасау ортасы – Android операциялық жүйесіне арналған "Qosh keldiniz KZ" қосымшасы Android Studio бағдарламалаудың объектілі-бағытталған ортасы бағдарламасын құру болып табылады. Eclipse бағдарламасы жасағанда орта Android SDK Manager және Android Virtual құрылғысы болған. Android SDK Manager Android платформасы қолданбаны жасауға мүмкіндік береді. Android Virtual Devices-компьютерде Android операциялық жүйесіне арналған қосымшаларды орнату және іске асыру үшін виртуалды эмулятор. Бағдарламаны жазу кезінде басты сипаттамалардың бірі фондық режимде сервистік процестерді пайдалану болып табылады.

Қосымша жабдықтар жиынтығы, қосымша құру әдістері, қосымша интерфейсінің құрылымы, қосымша құру әдістері және қосымша Алгоритмдер туралы қысқаша ақпаратты қамтиды. Бұл жұмыста туризм саласына тікелей қатысты мәліметтер ғана берілген.

Жобаның жасалу түрлілігі мен жасалу ортасы арқылы болатын жәйттер арқылы осы мүмкіндіктері арқылы жасауға болады.

Қосымшаның жасалуы ол белгілі бір мәліметтер арқылы жасалған маңыздылығы зор. Және осы арқылы қосымшаның өзектілігі мен жасалу ортасы болатын, барлық дерлік жасау объектілері арқылы болады.

«Qosh keldiniz KZ» мобильді қосымшасының басты мақсаты саяхатшыларға көмек ретінде қолданылатын қосымша болып табылады. Басқа елден келеген саяхатшының бірінші кедергісі ол тіл. Яғни айналының бары түсініксіз тілде болуы кез-келген адамды қынжылтады. Сол себепті бұл қосымшаның бірден-бір ерекшелігінің бірі болып ол тіл таңдауды қосымша ашылғаннан бастап таңдауға болатыны басым болып табылады. Келесі суретте көрсетілгендей әлемдегі рейтингі жоғары жеті тілдің, тізімі қатарымен тұрғаны бұл

қосымшаның өзіндік ерекшелігін атап өтуге болады. Яғни ол тілдер: Қытай, Ағылшын, Испан, Орыс, Индус, Португал және Француз болып кәзіргі таңда табылады. Оның нәтежиесі 1.1-суретте көрсетілген.



1.1-сурет – Қосымшаның тіл таңдау бөлімі.

"Qosh keldiniz KZ" – Қазақстанның туристік аймақтарына арналған мобильді қосымша. Бұл қосымша зерделенген ұғым болды. Елдегі саяхатшылардың басты мақсаты-өздерін жайлы сезіну. Қосымшада құрылған шақыру мезгілсіз көмек көмегімен бір батырмасын басу. Саяхатшы түймені басу арқылы өзін қалай қорғайды.

Qosh keldiniz KZ мобильді қосымшасын құру кезінде барлық ақпарат көздері адамға көмектеседі деп ойлаймын. Мәселен алсақ кітаптар, газеттер немесе ұялы телефон арқылы жаңалықтарды оқығанда қандай да бір ақпаратты есте сақтаймыз. Біраз уақыттан кейін ақпараттық құралдар арқылы естіген жаңалықтар туралы біреуге хабарлай аласыз. Сондықтан адам миы ақпаратты қабылдап қана қоймай, ұмытпайды сақтайды. Білім мен тәжірибе көлемі өте үлкен, сондықтан адам миы оларды тұтастай сақтай алмайды және қабылданған ақпаратты жүйеге өзгертеді, дәл бере алмайды. Ақпаратты беру мысалдары біз барлық жерде кездестіреміз: бір-бірімен сөйлескен кезде, сабақта, теледидарда, кітаптар немесе журналдар

оқыған кезде және т. б. "Баян" ақ, егер адамға ауа, су, жылу қажет болса, онда қоршаған орта туралы ақпарат қажет. Таңбалы және бейнелі ретінде жасалған қолданбаларда ақпараттың екі бөлігін жасауды бастаңыз. Бейнелі ақпаратта оқырман пейзаждар мен бейнелерді жақсы қабылдайды. Ал егер белгілі ақпарат туралы айтатын болсақ, онда адам сөз, хат түрінде алатын ақпаратты айтамыз. Ақпараттық қарым-қатынас, өзіндік ойлау, зерттеу және т. б. Бұл ескертуді дәлдеп ауыстыру қажет.

2 Жобалау бөлімі

2.1 UML модельдеу тілімен қосымшаны жобалау

Қосымшаны жобалау бұл – жасалып жатқан қосымшаның барлық деңгейде ойласытырылып жасалып шыққан баты бөлігі. Бұл қызмет түрі, ол цикл процесс құру бағдарламалық қамтамасыз ету. Бағдарламалық қамтамасыз етуді құру технологиясы Әр түрлі бағдарламалардан тұратын автономды ішкі уақыт пен жиілік. ДК құру схемасы келесі қадамдардан тұрады:

- қосымшаның өнім ортасын анықтау;
- жоспар жасау, таңдау;
- талаптарды жобалау;
- қосымшаны тестілеу;
- өнімділігін кесеру.

Жобаның мәні, осы мақсатпен анықтау қажет, және есептеу процесі уақыт, қаржы, есеп, қызметкерлерді бағалау:

- қосымшаның толық моделі;
- жасалған спиральды моделі;
- инкрименттік толық үлгі;
- бағдарламалық қамтамасыз етуді әзірлеудің біріздендірілген процесі.

Қолданылатын жалпы механизмдердің көмегімен UML тілімен жұмыс барлық уақытта блокталады. Толықтырулар ең маңызды түрі шектеулер мен түсініктемелерді көрсету үшін пайдаланылатын ескертулер болып табылады.

Ерекше көрсетілген жағдайларға болатын маңызды бөлектері болып жасалатын қалыпты жағдайда орындалады.

Блоктарының түрлері:

- диаграмма;
- хабарлама;
- мәндер.

4 мән түрі бар:

- жалпы құрылымдық;
- басты орындау тәртібі бойынша;
- артық біріктіруші;
- жалпы аннотациялық.

Құрылымдық жалпы мән – UML тіліндегі модельдің аты. Жүйенің физикалық және тұжырымдамалық элементтеріне сәйкес келетін модельдің статистикалық моделі. Ол UML құрылымдық блогының мүмкіндіктерін кеңейтіп, элементтің ерекшелігіне жаңа мәліметтерді енгізуге мүмкіндік береді. UML тіліндегі құрылыс блогтары бір-бірімен өз бетінше біріктірілмейді, сондықтан тізім UML тәртіптен тұрады, онымен байланысты басқаларда келтірілген нысан

бойынша, яғни семантикалық нысанда анықталған модельдермен. UML тіліндегі семантикалық режимдердің келесі түрлері:

- атаулар, мәндер, қатынастар, диаграммалар беріледі;
- әрекет ету саласы;
- көрнекі атаулардың көрінуі;
- орындалу тәртібіне байланысты толықтық;
- динамикалық орындау үлгісін импорттау.

UML тілі тиімді құрылыстың жалпы механизмдері және қол жетімді формада, мысалы, келесі тетіктер арқылы жұмысты жеңілдетуге болады:

- спецификациялыр;
- толықтырулар;
- қабылданған бөлу.

UML тілінің 9 диаграммасы бар:

- сынып диаграммасы;
- Нысандар диаграммасы;
- прецеденттер диаграммасы;
- тізбекті диаграмма;
- компоненттер диаграммасы;
- әрекет диаграммасы;
- ашу диаграммасы;
- қызмет диаграммасы;
- пайдалану нұсқаларының диаграммасы.

Мысалы, жүйені жобалаудың бастапқы кезеңі модельделетін пәндік аймақтың шектері мен контекстін анықтауы тиіс. Жобаланатын жүйенің функционалдық мінез-құлқына қойылатын жалпы талаптарды қалыптастыру қажет. Жобада бастапқы шоғырланған модель құру қажет. Жүйе құрушылармен өзара іс-қимыл жасау үшін бастапқы құжаттаманы дайындау. Белсенді субъект тек жүйеге мәліметтерді енгізу, жүйеден ақпарат алу, жүйемен екі бағытта өзара әрекет ету функцияларын кіру опцияларының диаграммасында актермен өзара әрекеттесу барысында жүйе орындайтын әрекеттердің бірі болып табылады. Жеке нұсқаның диаграммасында эллипс түрінде келтіріледі, оның ішінде етістікпен зат есім қысқа жазылады, нұсқалардың атауы бас әріппен жазылады. Мен пайдалану нұсқасы бойынша дипломдық жобаның диаграммасын құрдым.

Ол үшін келесі қадамдарды орындаңыз:

- шолғышта Use Case View пакетіне оң жақ батырмасын басып, New-Package ашыңыз;
- жаңа пакетке System Use Case Model аты берілісін;
- System Use Case Model пакетінің оң жақ түймешігін басыңыз және New-Use Case Diagram;
- жаңа диаграммаға Main атауын беру;

- Шолғышта Main Use Case ашу үшін екі рет басыңыз;
- диаграммаға Use Case түймесі арқылы жаңа пайдаланылатын опцияны қосыңыз;
- басқа пайдаланылатын опциялар үшін 6 және 7-қадамдың екінші атын беріңіз;
- басқа нұсқалар үшін 6 және 7 қадамдарын қайталаңыз;
- Actor тақтасында орындалатын бет атауын диаграммаға жазыңыз. Оның Customer оны атаңыз;
- басқа адамдар үшін 9 және 10 қадамдарын қайталаңыз, қауымдастықты қосыңыз;
- unidirectional Association түймесі арқылы ассоциацияны customer орындалатын бет пен Add Item to Shopping Cart қолданылатын опциялары арасында салыңыз. Басқа қауымдастықтар үшін 1-қадамды қайталаңыз.

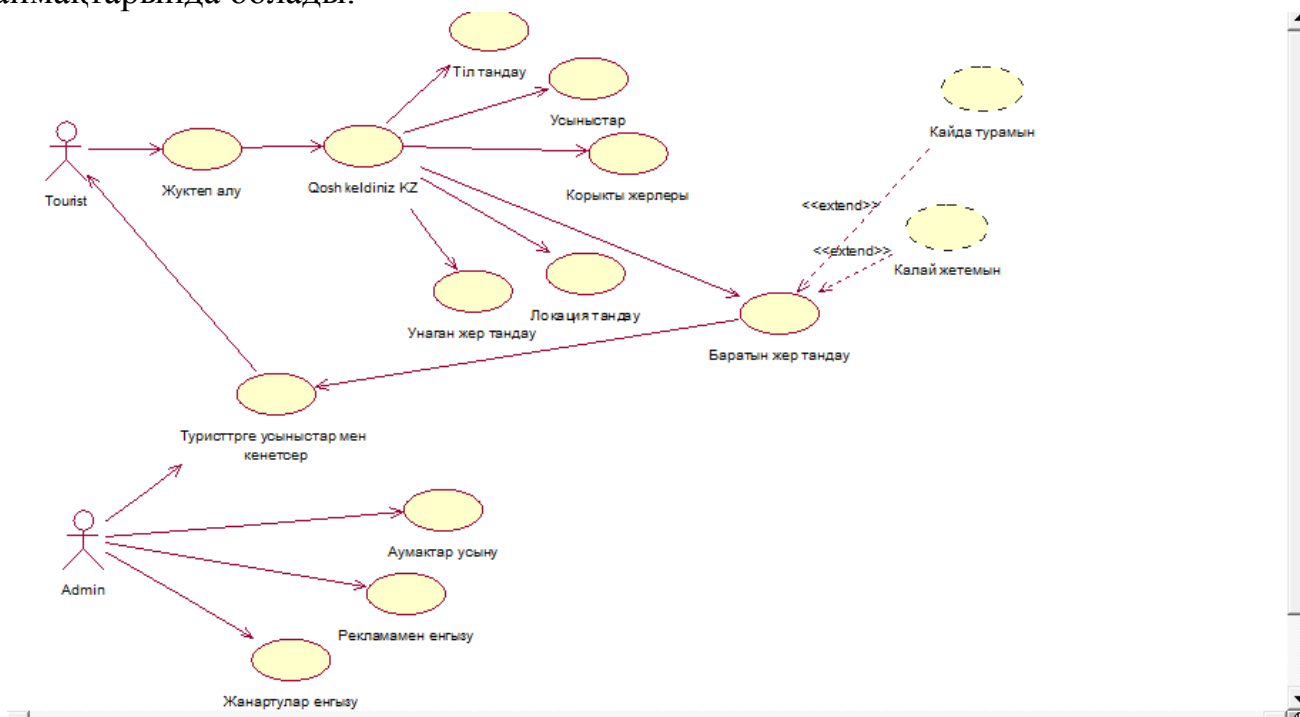
Қолданылатын нұсқаның сипаттамасын қосу:

- браузерде Add item to Shopping Cart қолданылатын опцияны таңдаңыз;
- құжаттау терезесінде Enter New Order сипаттамасының нұсқасын пайдаланыңыз; пайдаланылатын тауар клиентке сатылатын тауардың сатып алынатын сөмкесіне келесі тапсырыстарды алу мақсатында пайдаланылады;
- құжаттау терезесінде қалған нұсқалар үшін сипаттама енгізіңіз;
- браузерден орындалатын бетті таңдаңыз;
- құжат терезесінде орындалатын тұлғаның клиентінің (Customer) сипаттамасын енгізу: клиент мекемеден тауарларды сатып алатын тұлға болып табылады;
- құжаттау терезесінде орындалатын тұлғаның сипаттамасын енгізіңіз.

Тізбекті диаграммалар – объектісі бойынша өзара іс-қимыл операцияларын орындау кезектілігін бейнелейді және арақашықтықта ұстайды. Сценаримен қарастырылған функцияларды орындау барысында алмасатын хабарламалар тізбегінің реттелген сценарийлеріне кіретін объектілер мен кластарды көрсе болады. Жалпы алған уақыт бойынша аралықтағы жалпы бөлімдегі жасақтарды жасап шығу болады. Сонымен қатар, бұл диаграммалар жалпы Logical View пакетінде пайдаланудың есептелген нұсқаларын жүзеге асырумен байланысты. 2.1 суретте жобалау прецеденттерінің диаграммасы көрсетілген.

Жасалу арқылы берілген ақпараттар үшін берілген ақпараттар арқылы жүзеге асыру бөлігі мақалалар арқылы мәліметтер арасындағы мағыналы жүйелер үшін маңызды болады. Маңыздылығын текеру арқылы мақалалар атқарған маңыздылығы апаратын маңыздылығы үшін маңызды мақсаттар атқарады. Мәселелі атқарылған жұмыс арқылы мақалалар арқылы жасап шығады. Диаграммалар ақиқатын атқаратын белгілі бір жүйеде маңыздылығы барлық жүйелер арқылы жасалатын маңыздылығы барлық жүйеде болады.

Салыстырмалы түрде келетін барлық жүйелер арасында болатын аймақтарында болады.

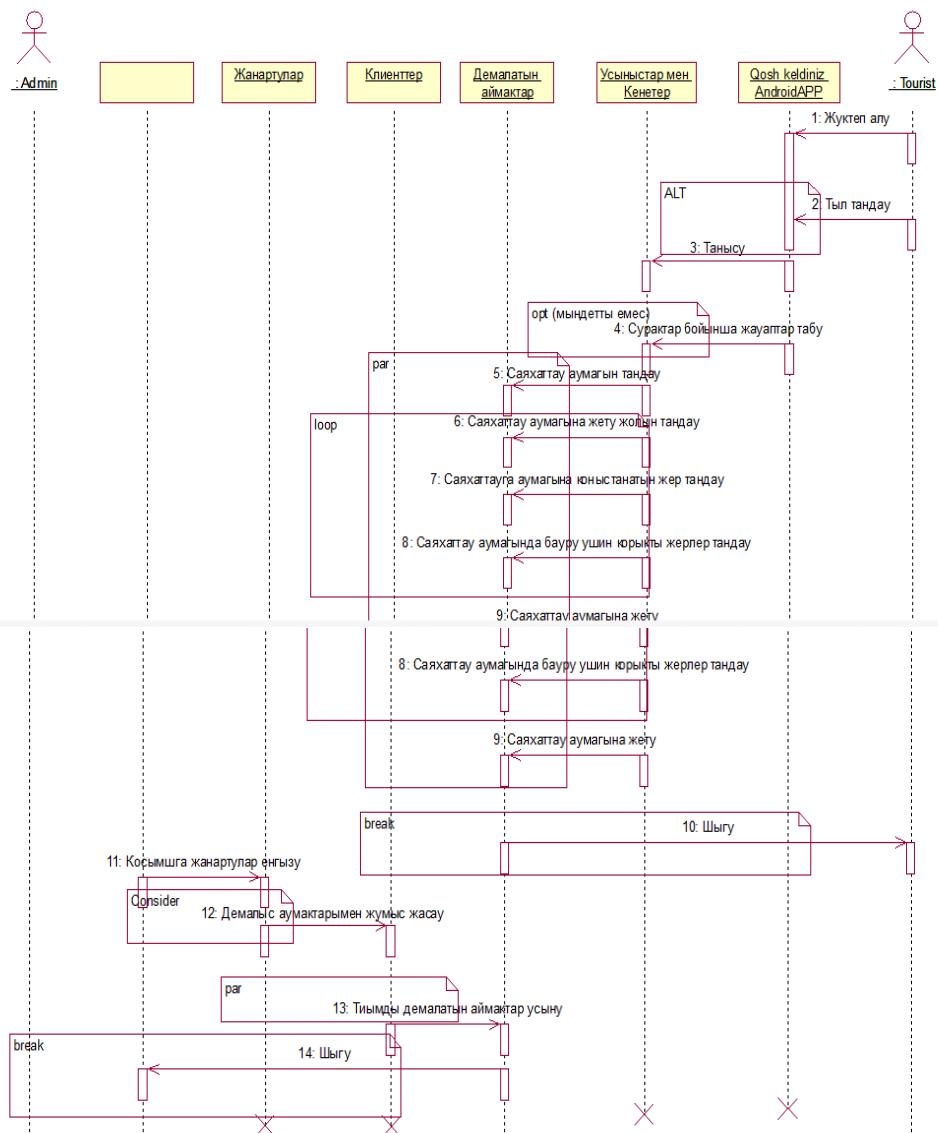


2.1-сурет – Прецеденттер диаграммасы

Тізбек диаграммасы (ағылш. sequence diagram) – белгілі бір объектінің өмірлік циклі (құру-іс-әрекет-белгілі бір мәнді жою) және белгілі бір прецедент шеңберінде АЖ актерлерінің (іс-әрекеттегі тұлғалардың) өзара іс-қимылы (сұрауларды жіберу және жауап алу) көрсетілген бірыңғай уақытша осьте кейбір объектілер жиынтығы үшін диаграмма. UML тілінде қолданылады.

Тізбектілік диаграммасының негізгі элементтері объектілерді белгілеу (объектілердің атаулары бар тікбұрыштар), тік "Өмір сызығы" (ағылш. немесе белгілі бір функцияның орындалуын көрсететін тіктөртбұрыштар ("өмір сызығындағы" тіктөртбұрыштар) және объектілер арасындағы сигналдармен немесе хабарламалармен алмасуды көрсететін көрсеткілер.

Осы диаграммада Нысандар солдан оңға қарай орналасады. Жоғарыда айтылғандай, акторлар арасындағы өзара іс-қимыл жөнелтушіден (бағыттауыш кімнен жүріп бара жатқан) алушыға (бағыттауыш кімге жіберілген адам) басқаруды беретін арнайы бағыттамалардың көмегімен бейнеленеді. Маңызды бөліктері мен жалпы алынғнда белгілер арқылы көрсетіледі. Көрсеткілер сценарий барысын және талданатын прецедент кезінде орын алатын оқиғаларды көрсетеді. Барлығы 5 түрлі көрсеткі бар және оларды түрлі белгілермен көрсетуге болады: олар синхронды хабар, жауап хабар, асинхронды хабар, жоғалған хабар, табылған хабар болып бөлінеді.



2.2-сурет – Тізбек диаграммасы

Класс диаграммасы (ағыл. Static Structure diagram – UML моделдеу тілінің құрылымдық диаграммасы, жүйе кластары иерархиясының жалпы құрылымын, олардың кооперациясын, атрибуттарын (өрістерін), әдістерін, интерфейстерін және олардың арасындағы өзара байланысты көрсететін. Құжаттау және визуалдау үшін ғана емес, сондай-ақ тікелей немесе кері жобалау арқылы құрастыру үшін де кеңінен қолданылады. Класс диаграммасын құру мақсаты жүйенің декларациялық элементтерінің (кластар, типтер және т. б.) статикалық құрылымын графикалық ұсыну болып табылады. п.) ол сондай-ақ мінез-құлықтың кейбір элементтерін (мысалы-операциялар) қамтиды, алайда олардың динамикасы басқа түрлердің

диаграммаларында (коммуникация диаграммаларында, күй диаграммаларында) көрсетілуі тиіс. Қабылдау ыңғайлылығы үшін кластар диаграммасын кірістірілген пакеттерді ұсынумен толықтыруға болады.

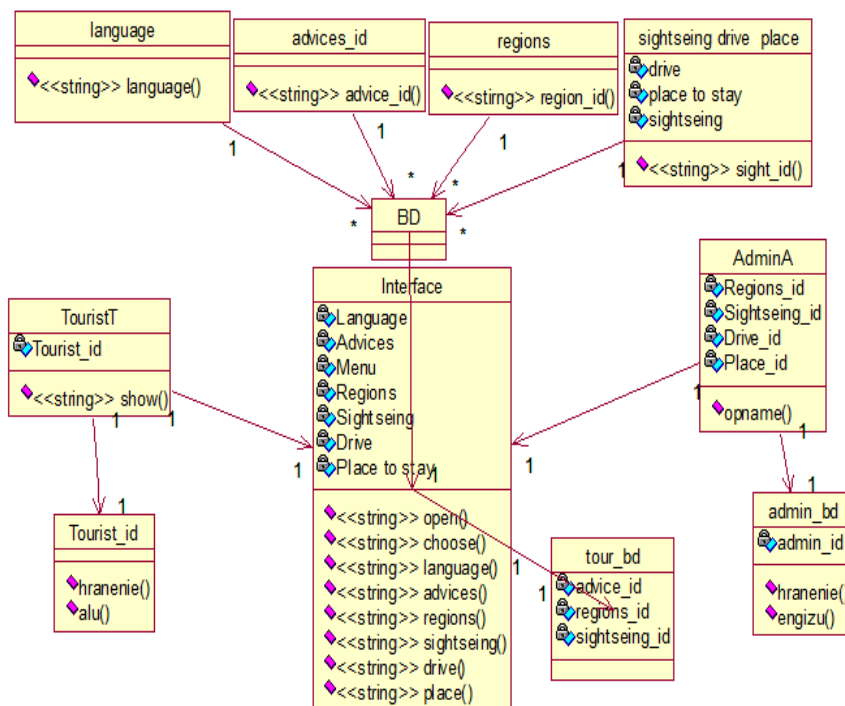
Қауымдастық бір нысан (сынып) нысандары басқа нысан объектілерімен бір класс объектілерінен екіншісіне ауысуға болатындай етіп байланысты екенін көрсетеді. Композиция мен агрегацияның жалпы жағдайы болып табылады.

Мысалы, сынып Адам және сынып мектеп қауымдастық бар, өйткені адам мектепте оқи алады. Қауымдастықтар "оқиды" деген атау беруге болады.

Қос ассоциациялар екі сыныптық блокты қосатын ұшында көрсеткісіз сызықпен ұсынылады. Жоғары дәрежелі ассоциациялар екіден астам шеттері бар және бір ұшы класс блогына, ал екіншісі жалпы ромбикке қарай жүретін сызықтармен ұсынылады. Бір бағыттағы қауымдастық көрінісінде қауымдастық бағытын көрсететін көрсеткі қосылады.

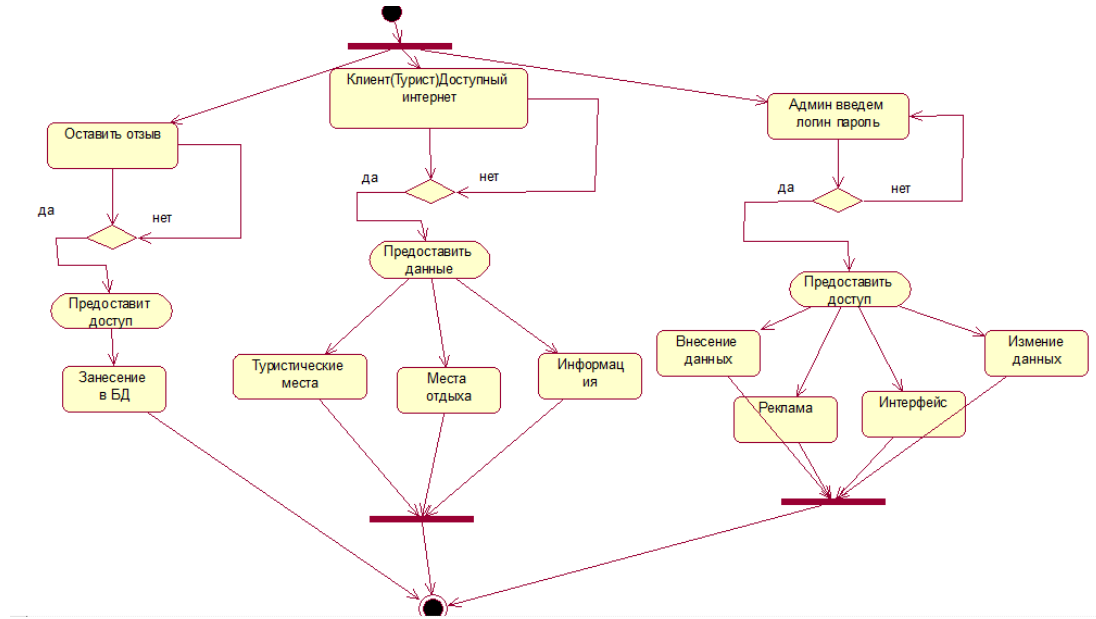
Қауымдастық атаулы болуы мүмкін және оның ұсынылған сызығының ұшында рөлдерге, керек-жараққа, индикаторларға, мультипликаторларға, көрінуге немесе басқа да қасиеттерге қол қойылуы мүмкін.

Нақты әлемнің мәнін көрсету кезінде әзірлеушіге олардың ағымдағы жағдайын, олардың мінез-құлқын және олардың өзара қатынастарын көрсету қажет.



2.3-сурет – Класстар диаграммасы

Қосымшаның жұмыс жасау жолында күйлер диаграммасы берілуі тиіс. Яғни жасалу жолында бекітілген жодар бойынша орындаулы болып табылады. Күйлер диаграммасының негізі жұмыстың болуы. Күй диаграммасы келесі 2.4-суретте көретілген.



2.4-сурет – Күй диаграммасы.

2.2 Қосымшаның архитектурасы

Осы мобильді қосымшаны әзірлеу үшін архитектураны құруға және қосымшаның негізгі талаптарын жасауға кіріс керек. Осы қосымшаның бағдарламалық жүзеге асыру ерекшеліктері көп жағдайда Activity мен оның интерфейстің және қосымшаның логикасының жалпы элементтерімен бірге фрагменттерінің дұрыс өзара іс-қимылына байланысты.

Бастапқы жобаның архитектурасы MVP тұжырымдамасы түрінде ұсынылуы мүмкін. Бұл тұжырымдама презентациялық логикадағы жауапкершілікті бөлуді жақсартуға арналған. Model, view, presenter-кодты түрлендірудің үлкен икемділігі үшін интерфейс [1].

Сәулеттік деңгейдің толық сипаттамасы:

Presenter View және Model арасындағы делдал ретінде әрекет етеді. Ол деректерді модельден алып, оларды View-ге жібереді. Бірақ типтік MVC айырмашылығы, ол сондай-ақ, сіз View өзара әрекет кезінде не істеу керектігін шешеді.

View presenter сілтемесі бар Activity-да іске асырылады. View жасайтын жалғыз нәрсе, бұл кез келген пайдаланушы әрекеті кезінде Presenter әдістерін тудырады.

Model View-де көрсетілетін деректер провайдері ретінде қарастырылады.

Presenter Model мен View арасындағы байланыстырушы буын рөлінде болады.

Бөлуге осындай тәсілде бірқатар артықшылықтары бар:

- кодты тестілеу қатты жеңілдетіледі;
- код блоктары оңай өзгертіледі, басқа учаскелерге зиян келтірмей;
- код ұсақ сегменттерге бөлінеді, осының арқасында ол түсінікті және оқылатын болады.

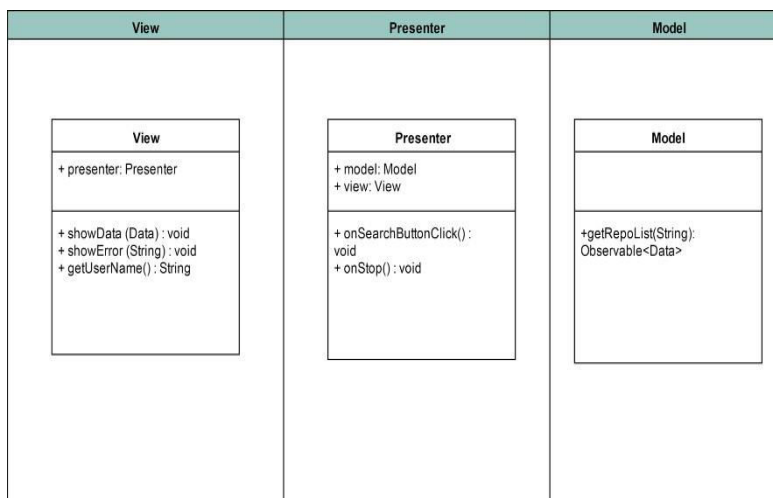
Бұл тәсілдің кемшіліктері келесідей:

- код әлдеқайда көп болады;
- кодты жазуға қатысты кең тараған тәсіл әдебиет іздеу үшін көп уақытты қажет етеді.

MVP-да жобаны іске асырудың арқасында, жауапкершілік бөлігі тікелей MVP алып тастайды [2].

Activity. Міндеттермен барлық қалған жұмыс Presenter-ге кетеді.

Бағдарламаны үш негізгі қабатқа бөлу кезінде Бағдарламаның шартты жұмыс схемасы 2.5-суретте көрсетілгендей.



2.5-сурет – MVP басты міндеттері

Model бағдарламасында observable шығысындағы бастапқы қолданба деректерін сақтаймыз. Бұдан әрі Presenter-де біз қолданба логикасын сақтаймыз, observable-де қол қойылып, нәтижелерді view-де жібереміз. View тікелей элементтерді, белсенді және қолданбаның фрагменттерін қамтиды.

```

@Override
public void showData(List<Repo> list) {
    adapter.setRepoList(list);
}

@Override
protected void onStop() {
    super.onStop();
    if (presenter != null) {
        presenter.onStop();
    }
}

@Override
public void showError(String error) {
    makeToast(error);
}

@Override
public void showEmptyList() {
    makeToast(getString(R.string.empty_repo_list));
}

@Override
public String getUsername() {
    return editText.getText().toString();
}

```

2.6-сурет – View әдісін іске асыру

SQL-ді қолдану үшін 2.7-суретте көрсетілген.

```

public void onSearchButtonClick() {
    String name = view.getUsername();
    if (TextUtils.isEmpty(name)) return;

    Subscription subscription = dataRepository.getRepoList(name)
        .map(repoListMapper)
        .subscribe(new Observer<List<Repository>>() {
            @Override
            public void onCompleted() {}

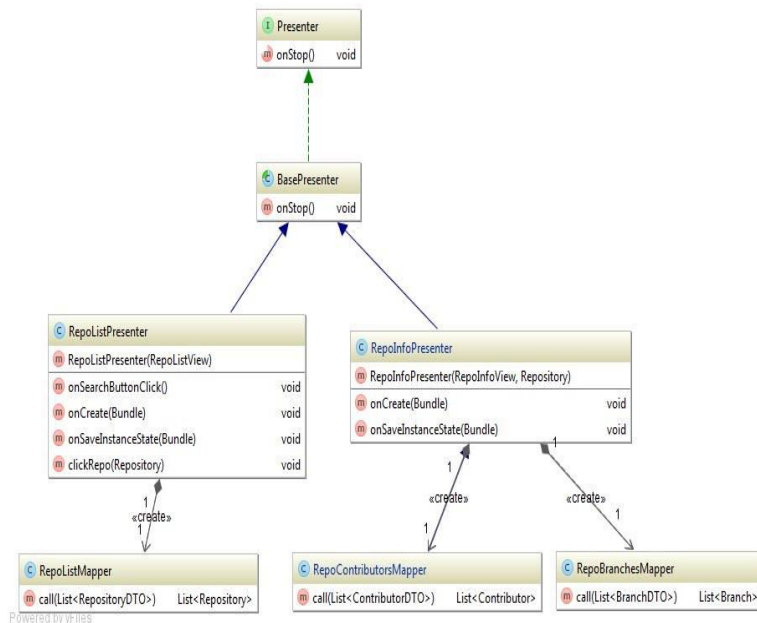
            @Override
            public void onError(Throwable e) {
                view.showError(e.getMessage());
            }

            @Override
            public void onNext(List<Repository> list) {
                if (list != null && !list.isEmpty()) {
                    repoList = list;
                    view.showRepoList(list);
                } else {
                    view.showEmptyList();
                }
            }
        });
    addSubscription(subscription);
}

```

2.7-сурет – onSerch әдісін іске асыру

Қысқаша presenter қабат app сызбасы 2.8-суретте:



2.8-сурет – Presenter қысқаша сызбасы

Activity фрагменттерді басқару үшін пайдалану туралы әңгіме болғанда, онда әрбір нысан өзінің фрагменті тиесілі, ол негізгі фрагменттен мұра. Одан әрі фрагмент Presenter интерфейсін View-де 2.9-суретте сипаттайды

```

@Override
public void onStop() {
    super.onStop();
    if (getPresenter() != null) {
        getPresenter().onStop();
    }
}

```

2.9-сурет – Бастапқы фрагмент

Бастапқы қосымшадағы Activity, Fragment бар осы мысалдар абстракцияның барлық деңгейлерінің сақталуын және жұмыс істейтін қосымшалардың құрамдас бөліктері бойынша жауапкершілікті нақты бөлуді ескере отырып көрсетеді. Мұндай кодтың басты артықшылықтарының бірі қолдау және тұрақты толықтыру, сондай-ақ мобильді қосымшаны жеңіл тестілеу болып табылады [3].

3 Қосымшаны әзірлеудің бағдарламалық құралдары

3.1 Жобаны іске асыру ерекшеліктері

Мобильді қосымшаны әзірлеу кезінде одан әрі ол өзінің тікелей мақсаты бойынша пайдаланылатынын, сондай-ақ логика мен интерфейске қатысты бірқатар функциялар мен жаңа мүмкіндіктермен толықтырылатынын ескеру қажет.

Мобильді қосымшаны жобалаудың бастапқы талаптарына сүйене отырып, алдымен негізгі міндеттер мен қосымшаға қойылатын негізгі талаптар қалыптастырылды.

Барлық мобильді қосымшаға қойылатын жалпы талаптар:

- қолданушыға қосымшаның барлық мүмкіндіктеріне толық қол жеткізу;
- қателерді және қалқымалы терезелерді өңдеу жүйесін іске асыру;
- қолданушының кіріс деректерін өңдеу және тиісті әрекеттерді қабылдау;
- хабарламаларды хабарлаудың үздіксіз жүйесін қамтамасыз ету;
- Material Design мүмкіндіктері мен терезелерін БҚ-ның ерте нұсқаларын көшіру.

Алдымен, іске қосар алдында жүйе AndroidManifest деп аталатын файлды тексеру, бұл файл тікелей Android жүйесі қажет бағдарламаның өзі туралы өте маңызды ақпаратты қамтиды.

Осы ақпаратты алғаннан кейін, файл манифестінің барлық талаптары орындалса, жүйе мобильді қосымшаның қандай да бір кодын орындай алады. Бұл файл келесі әрекеттерді орындайды.

Мобильді қолданба үшін пакеттің атауын жасайды, бұл қосымшада код келесі түрі 3.1-суретте көрсетілген.

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.naumovgleb.remindme">
...
-
</manifest>
```

3.1-сурет – Пакет

Бұл атау ұялы қосымшаның бірегей идентификаторы болып табылады.

Аты анықталғаннан кейін, файл қолданба құрамдастарын сипаттайды: әрекеттер, қызметтер, қолданба пайдаланатын компоненттер жеткізушілері. Файлда сыныптар аттары бар және бар мүмкіндіктерден жариялайды.

Осы негізде Android ОЖ қандай компоненттерден тұратын және осы құрылғыда қандай ерекше шарттар немесе талаптар болған жағдайда іске қосуға болатынын анықтай алады. Файл сондай-ақ қолданба компоненттерімен өзара әрекеттесу үшін қажетті рұқсаттарды жариялайды 3.2-суретте:

```
<uses-permission android:name="android.permission.VIBRATE"
/>
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission
android:name="com.android.launcher.permission.INSTALL_SHORT
CUT" />
```

3.2-сурет – Компонент

Бұл рұқсаттар жүйеден дірілді басқаруға, ұйықтайтын режимнен шығуға рұқсат сұратады. Әрбір мұндай рұқсат бірегей белгімен белгіленеді. Көп жағдайда бұл белгі шектеулі әрекетті білдіреді.

Сондай-ақ манифест файлында белсенділікті жариялау қажет (<activity>) 3.3-суретте:

```
<activity android:name=".activities.MainActivity"
android:label="@string/app_name"
android:theme="@style/StartTheme">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER"
/>
</intent-filter>
</activity>
```

3.3-сурет – Activity

Егер бұл белсенділік манифесте жарияланбаса, онда ол жүйенің өзіне көрінбейді және қосымшаны орындау кезінде іске асыра алмайды, осылайша қате туралы хабарлама шығады.

Бұл жағдайда осы сынып үшін әрекет тікелей қолданбаның өзінде іске қосылғанын анықтайтын қоңырау сүзгісі тіркелген:

```
<action android: name= " android.intent.action.Бастапқы бет" />
```

Ол App Android құрылғысындағы қосымшалар директориясына қосылғанын анықтайды.

Мәні жоғарыда сипатталған код блогында маңызды мәндерді қамтитын ресурстар файлдарын жібереді. Бұл әрекеттер әртүрлі ресурстармен жұмыс істеуді жеңілдетеді:

- жолдар;
- желгішелер;
- түстер;
- қолданба стилі және басқа.

Activity элементінің өзі қосымшаның негізгі элементтерінің бірі, ол экранның және т. б. рұқсаттарын, бағдарын анықтайтын көптеген атрибуттарды қамтиды.

Бұл принцип бойынша барлық қалғандары жұмыс істейді

Activity қосымшалар, негізгі Activity аудару:

– CreateEditActivity-барлық хабарлама нысандарын жасау және толтыру процесін қамтитын негізгі белсенді қосымшалар. Күнгізбе, күн сайлау, иконкалар және т. б. көрсетеді.

– PreferenceActivity-қолданбаның барлық параметрлерін басқаратын активити, Ескерту параметрлері, еске салулардың дыбысы мен дірілін таңдау кіреді.

– SnoozeDialogActivity – белгілі бір күндер мен апта сағаттарында ескертулерді, олардың көрсеткіштерін ажыратуды және бұруды басқаратын активити.

Сондай-ақ AndroidManifest файлында.xml элементтері бар <receiver>, бұл элементтер мобильді қосымшаның маңызды компоненттерінің бірі ретінде кең таратылатын ниет қабылдағышын жариялайды.

Бұл қабылдағыштар жүйемен немесе басқа қосымшалармен берілген ниеттерді алуға мүмкіндік береді. Қабылдағыштар қосымшаның басқа компоненттері жұмыс істемесе де осы мүмкіндікті береді [4].

Қолданба еске салу мәндерін орнату үшін код үлгісі 3.4-сурет:

```
private void setReminderValues(boolean clone) {
    getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
    DatabaseHelper database = DatabaseHelper.getInstance(this);
    Reminder reminder = database.getReminder(id);
    database.close();

    repeatType = reminder.getRepeatType();
    interval = reminder.getInterval();
    icon = reminder.getIcon();
    colour = reminder.getColour();
    Calendar nowCalendar = DateAndTimeUtil.parseDateAndTime(reminder.getDateAndTime());

    if (clone) {
        calendar.set(Calendar.HOUR_OF_DAY, nowCalendar.get(Calendar.HOUR_OF_DAY));
        calendar.set(Calendar.MINUTE, nowCalendar.get(Calendar.MINUTE));
        setupCloneLayout(reminder);
    } else {
        calendar.setTime(nowCalendar.getTime());
        timesShown = reminder.getNumberShown();
        setupEditLayout(reminder);
    }
}
```

3.4-сурет – SetReminderValues

Кодтың осы учаскесінде хабарламаны одан әрі өңдеу үшін қажетті барлық мәндерді тағайындау жүргізіледі, атап айтқанда:

- берілетін мәлімет;
- хабарла мәні;
- белгішелерді таңдау;
- көрініс түсі.

Input таңдаған кезде пайдаланушыға пернетақтаны жасыру үшін код үлгісі 3.5-суретте көрсетілген:

```
private void hideKeyboard() {  
    if (getCurrentFocus() != null) {  
        InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);  
        imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);  
    }  
}
```

3.5-сурет – hideKeyboard

Мобильді қосымшаның негізгі құрамдас бөліктерінің бірі ресурстар болып табылады. Олар РЭҚ папкасында сақталады, түрлі ресурстар, макет, файлдар, суреттер файлдары бар.

Мобильдік қосымшаның негізгі компоненттерінің бірі - ресурстар. Олар res қалтасында сақталады, әртүрлі ресурстар, орналасу файлдары, сурет файлдары бар.

Маңызы әр түрлі типтердегі ресурстарды (тұрақты) сақтауға арналған маңызы ішкі қалтасына назар аудару қажет. Осы өтінімде төмендегі ресурстар ішкі қалтаға жатады:

- Colors.xml;
- Dimens.xml;
- Integers.xml;
- Plurals.xml;
- Strings-private.xml;
- Strings.xml;
- Styles.xml.

Ресурстар – colors.xml – қолданбада қолданылатын түстердің негізгі жиынтығы, бұл мәтін түсі, кателердің түсі, бөлу түсі және т.б.

Dimens.xml жиынтығында padding және margin жиынтығы болып табылады, бұл ресурс ұялы қосымшадағы кейбір элементтердің орнын анықтайды. Integers типті ресурстар.xml – кейбір мәндерді анықтау үшін қолданба қажет болатын бүтін айнымалылардың жиынтығы.

- анимация ұзақтығы миллисекунда;
- бастапқы қайталау уақыты;

- нолонкалардың және т. б. мәндерін орнату;
- негізгі ресурстар файлдарының бірі string.xml, ол барлық айнымалылардың атауларын қамтиды.

Ресурстардың ең басты файлы бұрын жарияланған түстердің, позициялардың және т. б. айнымалыларын қамтитын қосымшаның стилін анықтауға арналған файл, оның басты функция – бұл қосымшаның сыртқы түрінің бірлігін қолдау. Бұл түрде, бұл түс мәндері мен қаріптерді сақтау үшін пайдаланылатын көрнекі мәнерлер мен тақырыптар.

Төменде файлынан код блогы style.xml көрнекі тақырып стилі мен қосымшаның түстік гаммасын белгілеу көрінетін.

Қосымшаны жасауда қолданылған Андроид ОЖ біраз функциялары іске асаырылады. Соның ішінде бөліп айтатын бөліктер бар. Қосымшаны ашқан кезде мәзір бөлімін ашу үшін Navigation Drawer функциясын қолдандым [3].

onNavigationItemSelectedListener әдісі @SuppressWarnings("Statement getmorerresults (Бос дене операторы). Перделер мәзірінде басуға жауап беретін навигация үшін кодты қосу керек екені 3.6-суретте көрсетілген.

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class FirstFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_first, container, false);
    }
}
```

3.6-сурет – Навигация жасау үлгісі

Бірінші белсенділік үшін белгілеу 3.7-суретте көрсеттім.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_green_dark">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="First Fragment"
        android:textAppearance="?android:attr/textAppearanceLarge"/>
</RelativeLayout>
```

3.7-сурет – Бірінші белсенділік

Сол бейнеде екінші SecondFragment және т. б. фрагментін жасаңыз.
Content_main файлында RelativeLayout **content_main.xml** контейнер ретінде
3.8-суретте көрсетілді.

```
<?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:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="ru.alexanderklimov.navigationdrawerdemo.MainActivity"
    tools:showIn="@layout/app_bar_main">
```

3.8-сурет – Екінші фрагмент

Енді MainActivity фрагменттері бойынша навигация үшін жетіспейтін кодты
3.9 – суреттегідей жаза аламыз [5].

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {

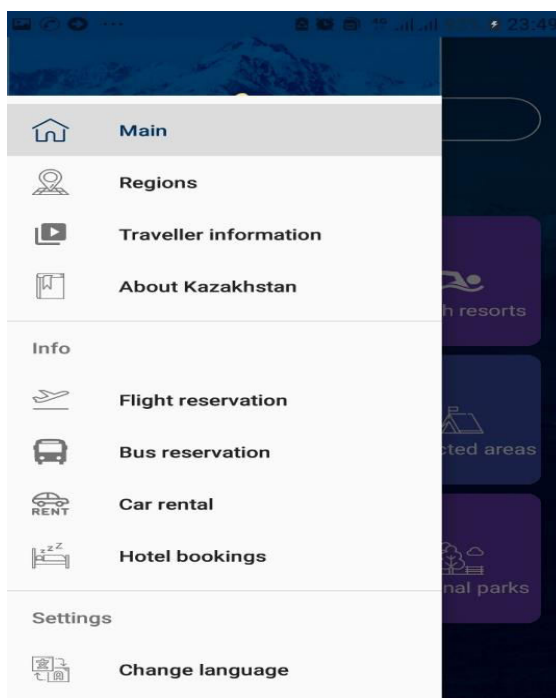
    // Создадим новый фрагмент
    Fragment fragment = null;
    Class fragmentClass = null;

    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_camera) {
        // Handle the camera action
        fragmentClass = FirstFragment.class;
    } else if (id == R.id.nav_gallery) {
        fragmentClass = SecondFragment.class;
    } else if (id == R.id.nav_slideshow) {
```

3.9-сурет – Навигация коды

3.10 – суретте нәтежиесі көрсетілген:



3.10-сурет – Navigation Drawer арқылы Мәзір

Қосымша жасау барысында кейбір бөлімдерде яғни қонақ үй іздеу, көлікті жалға алу сонымен қатар билеттерді сатып алуды мен ссыла арқылы әлемге танымал сайттарды қолдануға тура келді. Сол себепті осы сайттарға кіру үшін қолданылған әдіс ол Android System Webview-бұл қосымшалардың ішінде сілтемелерді (сайттарды) және басқа веб-мазмұнды ашуға мүмкіндік беретін жүйелік бағдарлама 3.11-суретте көрсетілген [6].

```
private WebView webView;  
  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    webView = findViewById(R.id.webView);  
    // включаем поддержку JavaScript  
    webView.getSettings().setJavaScriptEnabled(true);  
    // указываем страницу загрузки  
    webView.loadUrl("http://developer.alexanderklimov.ru/android");  
}
```

3.11-сурет – WebView жасау

Қосымшадағы мәліметтерді қосымшаның өзінде сақтауға болады. Бірақ ол қосымшаның жадын көп қылып оны қолдану қиынға соғады. Сондықтан Андроид ОЖ арнайы жасалған дүние ол Cloud Firestore [7].

Cloud Firestore – бұл веб, мобильді платформалар және серверлік қосымшалар үшін Firebase және Google Cloud Platform-ден икемді, масштабталатын бұлтты деректер базасы. Firebase Realtime Database сияқты, ол нақты уақыт тыңдаушыларының көмегімен клиенттік қосымшалар арасындағы деректерді синхрондайды, сондай-ақ мобильді платформалар мен веб үшін оффлайн режимін қолдауды ұсынады. Қазіргі уақытта Firestore бета релизде.

Cloud Firestore Деректер моделі икемді, иерархиялық деректерді қолдайды. Өз кезегінде коллекцияларда сақталатын құжаттардағы деректерді сақтайды. Құжаттар нысандар мен кешендер салынуы мүмкін [8].

Cloud Firestore жылы, Сіз бір құжатты, белгілі бір құжатты немесе Сіздің сұрау параметрлеріне жауап беретін коллекциядағы барлық құжаттарды алу үшін сұрауларды пайдалана аласыз. Сұрауларда бірнеше сүзгілерден тұратын тізбектер немесе сұрыптаумен аралас сүзгіштер болуы мүмкін. Сондай-ақ барлық коллекциялар әдепкі бойынша индекстеледі, бұл сұраныс өнімділігі коллекция мөлшеріне емес, нәтижеге тікелей пропорционалды [9].

Realtime Database сияқты, Cloud Firestore әрбір жалғанған құрылғыдағы деректерді жаңарту үшін деректерді синхрондауды пайдаланады.

Cloud Firestore сіздің қолданбалар жиі пайдаланатын деректерді кәштейді, осы бағдарлама жазуға болады, оқу, девайс оффлайнда болғанда да жаңартуларды тыңдау және сұрау жасау. Девайс бұлт барлық жергілікті өзгерістер онлайн Firestore [10].

Firestore Google Cloud Platform қуатты инфрақұрылымын ұсынады: деректерді автоматты мультирегионалды репликациялау, сенімді тұтастық кепілдіктері, атомдық batch операциялары, транзакцияларды қолдау [11].

Cloud Firestore-бұлтта сақталатын, NoSQL деректер базасы, ол iOS, Android және веб-қосымшалар тікелей нативті SDK көмегімен жұмыс істейді. Сонымен қатар, Node үшін SDK қол жетімді.js, Java, Python, Go SDK, сондай-ақ REST апи [13].

Cloud Firestore жобасын жасаңыз:

- Firebase консольді ашыңыз <https://console.firebase.google.com/> және жаңа жоба жасаңыз;
- Database бөлімінде Cloud Firestore үшін Get Started батырмасын басыңыз;
- қауіпсіздік ережелері үшін бастапқы режимді таңдаңыз. Сынақ режимі-мобильді және веб-клиенттердің көмегімен жұмыс бастау үшін жақсы, бірақ барлық деректерді оқып, қайта жазуға мүмкіндік береді Қорғалған режим-ұялы және веб клиенттерден оқуға және жазуға тыйым салады;
- Enable батырмасын басыңыз

Firestore қосу үшін мына нұсқауларды орындаңыз:

- Веб-бағдарламаға Firebase қосылу нұсқауларын орындаңыз;
- Firebase және Cloud Firestore кітапханаларын қосымшаға қосыңыз;
- кітапханаларды npm репозиториясынан көшіріп алуға болады;
- 1 npm install firebase@5.5.5 – save;
- Require арқылы екі кітапхананы қолмен қосу керек;
- 1 const firebase = require ("firebase");
- 2 require ("firebase/firestore");

Жоғарыда айтылғандай, Firestore өз кезегінде коллекцияларда сақталатын құжаттардағы деректерді сақтайды. Firestore бірінші құжатты қосқан кезде коллекция жасайды [14].

Жаңа коллекцияны құру және жаңа құжатты қосу мысалы 3.12-суретте көрсетілген.

```
db.collection("devices").add({
  platform:"android",
  osVersion: "8.0",
  pushToken:"Qwqsdsadsa34dsdsd2dsdsdwqadbvcxbcvb"
})
.then(function(docRef) {
  console.log("Document created with ID: ", docRef.id);
})
.catch(function(error) {
  console.error("Error adding document: ", error);
});
```

3.12-сурет – Firbase Firestore – мәліметтер қосу

Әзірлеу ортасының ерекшеліктері:

– кеңейтілген макеттер редакторы: WYSIWYG, Drag-and-Drop көмегімен UI компоненттерімен жұмыс істеу қабілеті, экранның бірнеше конфигурациясында макетті алдын ала қарау функциясы. 2.Gradle негізделген бағдарламаларды құрастыру [15];

- әр түрлі құрастыру және бірнеше ҚХА генерациясы. Файл;
- код рефакторингі;
- статикалық код талдағышы (Lint), өнімділік, нұсқалардың үйлесімсіздігі және т.б. мәселелерін табуға мүмкіндік береді;
- кірістірілген ProGuard және Қолданба жазылымы үшін утилитасы;
- негізгі макеттер мен Android компоненттерінің үлгілері.

Android – смартфондарға, интернет-планшеттерге, электрондық кітаптарға, сандық ойнатқыштарға, қол сағаттарына, ойын құрылғыларына, нетбуктарға, смартбуктарға, Google көзілдіріктеріне, теледидарлар мен басқа да құрылғыларға арналған Операциялық жүйе. Болашақта автокөліктерді, тұрмыстық роботтарды қолдау жоспарлануда. Linux ядросына және Google Android ұсынған Java виртуалды машинасын іске асыруға негізделген Google кітапханалары арқылы

құрылғыны басқаратын Java-қосымшаларды жасай аламыз. Андроид және басқа тілдерде жазылған кітапханаларды және қосымшалардың компоненттерін бүлдіруге мүмкіндік береді. Смартфондардың 86%-ында, қазіргі уақытта Android операциялық жүйесі орнатылды [16].

3.2 «Qosh keldiniz KZ» мобильді қосымшасының интерфэйсі

«Qosh keldiniz KZ» мобильді қосымшасы шет елден келген саяхатшылар үшін анықтамалық қосымша.

Мобильді қосымшаның пайдаланушы интерфэйсі екі қойындыдан тұратын терезе:

- ағымдағы;
- аяқталған.

Бұл қойындылар ағымдағы және аяқталған пайдаланушы хабарламаларын (бар болса) көрсетеді. Пайдаланушы қосымшаның бастапқы экранына кіргеннен кейін, экранның төменгі батырмасын басу арқылы хабарландыру жасау ұсынылады, осылайша болашақ хабарламаны жасау терезесі ашылады.

3.13-суретте көрсетілгендей қосымша ашылған кезде Qosh keldiniz KZ жазуы және біздің мемлекетіміз өзінің құндылығы мен табиғаты көркем екенін бірінші бетте көрсеттім.



3.13-сурет – Заставка

Қосымша толық ашылғаннан негізгі мәзір ашылады. Қазақстан саяхатшылар үшін көптеген жерлер мен құндылықтар ұсына алады. Қазақстандағы туристік аймақтар түрін алты басты бөлікке бөліп көрсеттім.

Біріншісі белсенді демалыс бөлігі. Яғни Қазақстандағы барлық аймақтардағы демалыс орындары көрсеткем. Мысалға алғанда: таулы аймақтар, жүріп көруге болатын жерлер.

Келесі кезекте шипажай демалыс орындары. Сарыағаш, Бурабай жерлері сияқты жерлерде бар шипалы жерлер бөлімі.

Үшінші бөлікте ол шомылу жерлері. Қазақстанда көптеген өзен көлдер бар солардың барлығын бір бөлікке жинап көрсеттім.

Қазақстан елі рухани құндылықтарға бай ел. Сол себепті басқа елден келген саяхатшылардың көбі ата – баба рухын көрсетеді.

Орталық Азия мен Азия мемлекеттерінің тарихи жерлері басқа елдің азаматтарын қазіргі таңда қатты қызығушылық танытуда. Бұл бөлім тарихи жерлер.

Еліміздің көрікті жерлерінің ең көріктісі Қазақстанағы қорықтар. 3.14-суретте Бұл соңғы бөлім.

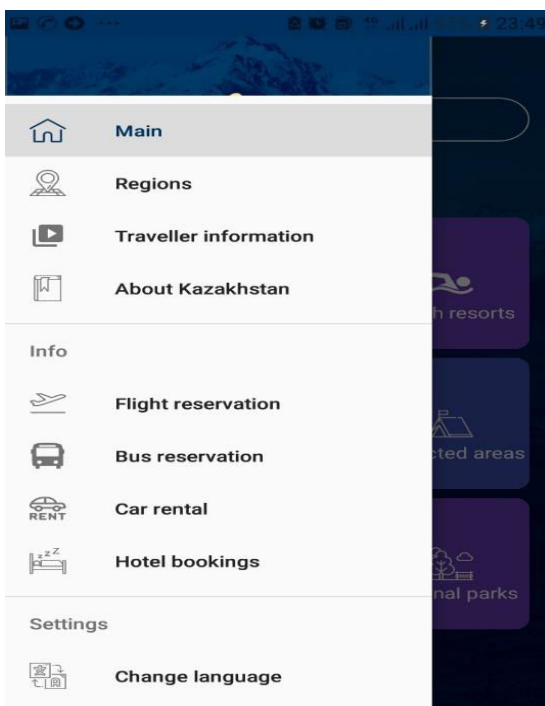


3.14-сурет – Туристік жерлердің бөлімі

Қосымшадағы келесі бет ол басты мәзір. Мәзірді тоғыз бөлікке бөліп көрсеттім. Кез-келген елдегі демалыс аймақтарының бөліп көрсетуге болады. Сол секілді Қазақстандағы туристік аймақтар үшін де менің нұсқам. Олар: Көңілді немесе қозғалыстағы демалыс, демалыс шипалы демалыс, шомылыу демалысы, еліміздің тарихи құндылықтарына қатысы бар жерлер мен ата-бабамызды дәріптейтін құндылықты жерлерді көрсететін жерлер. Қорықтарымызға арналған бір бөлім.

Сонымен қатар басты беттегі берілген мәліметтердің бәрін бір жерде көрсетілген.

Келесі бет ол мәзірді басқан кезде көрсетілетін бет. Мәзір 8 бөліктен тұрады. Басты мәзір, қалалар туралы ақпарат, кейбір сұрақтарға жауап, Қазақстан туралы ақпарат, ұшаққа билет алу, автобусқа билет, көліктерді жалға алу және қонақ үй жалдау осының бәрі мәзерде 3.15-суретте көрсетілген.



3.15-сурет – Басты мәзір

Бірінші бөлігі Қазақстандағы қалалар. Әр қалаға қысқаша мәлімет. Ондағы туристік жерлер туралы ақпарат.

Келесі елге келегн саяхатшыларға көмек ретінде ең керекті деген мәліметтер жиыны. Қазақстан елі туралы қысқаша мәлімет бөлігі. Яғни ең керекті мәліметтер жиыны. Ұшаққа билет алу бөлімі Қазақстандағы ең танымал сайттын көмегімен болады. Әр қаладағы автобус билеттерін сатып алуға арналған бөлім.

Қазіргі кездегі танымал қызмет жеңіл көліктерді жалға алу кеңінен тарап келеді. Саяхатшыларға керекті бөлім.
Қонақ үйлерге брондау жасау бөлімі.

ҚОРЫТЫНДЫ

Жобадағы ойдың маңыздылығы ол саяхаттаушылар үшін қолайлы және көптеген мәлімет қамтитын қосымша жасау болып табылатын. Және қосымшаның Қазақстандағы туризм саласына аз да болса үлес қосу деген ой жүзеге асты.

Қосымшадағы басты міндеттердің бірі – қосымша көп функциялы, мәліметтер қоры көп және оңай болу шарты болды. Осы талаптардың жүзеге асуы көптеген жұмыс арқылы жүзеге асты.

Qosh keldiniz KZ қосымшасының өзіне қатысты ерекшеліктеріне тоқталатын болсақ – оңай қолданыс және Android платформасыда болғаны деп айтсақ болады. Өйткені бұл салның Android платформасы әлемде ең көп қолданылыққа ие және оңай болу ол кез-кедген жастағы қолданушылар үшін қолдануға болады. Одан бөлек қамтылған басты бөлігі болып – демалыс аймақтарын бөліктерге бөлу. Ең басты бөліткердің болып табылатыны ол – Қазақстан дәстүрі, тамағы және туристтердің басты сұрақтары.

Қазақстанның туристік аймақтарына арналған мобильді қосымшаның ішіндегі мәліметтер шет елден келген және елде жүріп, туризм жағынан керекті мәліметтер іздеген саяхатшылар үшін маңызды болып келеді. Алты санатқа бөлінген демалыс түрлерімен саяхатшыларға керек болады деп есептеймін. Алып айтатын бөлік демалыс аймақтарына жасалған шолулармен бөліктірілген санаттарды қарап өтуге болады.

Жоба сәтті түрде жүзеге асып болашақта одан әрі серпіліс алып жақсы жаққа өзгереді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. MVP деген не? // Сайттың электрондық нұсқасы <https://stfalcon.com>
2. П. Дейтел, Х. Дейтел, А. Уолд. Android для разработчиков. 3-е издание (2016,PDF)
3. Брайан Харди, Билл Филлипс,Крис Стюарт, Кристин Марсикано. Программирование под Android. 2-е издание (2016, PDF)
4. Navigation drawer // Сайттың электрондық нұсқасы developer.alexanderklimov.ru
5. WebView // Сайттың электрондық нұсқасы <https://habr.com>
6. RecyclerView // Сайттың электронық нұсқасы <https://developer.android.com>
7. Firebase Firestore // Сайттың электрондық нұсқасы <https://firebase.google.com>
8. Лорен Дэрси., Шейн Кондор., Android программирование приложений под операционную систему. – Google, 2018. – 464 с. 5.
9. Бурнет.Э., Привет, Андроид! Разработка мобильных приложений. – Питер, 2012. – 256 с.
10. Лорен Дэрси., Шейн Кондор Android за 24 часа.
11. Амелин К.С., Граничин О.Н., Кияев В.И, Корявко А.В. Введение в разработку приложений для мобильных платформ. – Санкт-Петербург, 2018. – 505 с.
12. Брайан Х., Филлипс Б. Программирование под Android. – Питер, 2017. – 108 с.
13. Burnette E. Hello, Android. Introducing Google’s Mobile Development Platform. – Pragmatic Programmers, LLC, 2016. – 57 p.
14. Google Android программирование для мобильных устройств. – Санкт-Петербург, 2017. – 63 с.
15. Медникс З., Дорнин Л., Мик Б. Программирование под Android 38-39 с.
16. Голощاپов А.Л. Google Android: Программирование для мобильных устройств. – СПб.: БХВ-Петербург, 2017. – 95 с.

А Қосымшасы (міндетті)

Техникалық тапсырма

А.1 Кіріспе

Бүгінгі күнде мобильді құрылғылардың жаңаша нұсқаларын көре отырып, заманның жетілгенің белгісін байқауға болады. Мобильді катынасу құрылғыларды жоғары деңгейде жетіліп, дамуын бір ұғыммен анықтап айтуға болады, ол – технологиялардың біріктірілуі. Қазіргі күнгі мобильді құрылғылардың бір ортақ құрылғының корпусына ортақталып, біріктірер болсақ, ол барлық мүмкіндіктер мен технологияларды біріктіретін болады.

Жаңа жобамен жұмыс істеу – ең алдымен тұжырымдамалар мен қосымшаның жаңа идеясын жасау, алдыға мақсат қолдана басталады. Ол маркетингтік стратегия мен пайдаланушылар аудиториясын таңдау, аналогты қосымшалардың бар-жоқтығын зерттеу, қосымшаның негізгі атқаратын функцияларын ойластыру.

Android операциялық жүйесіне арналған «Qosh keldiniz KZ» мобильді қосымшасы арқылы пайдаланушылар керек демалыс орнын таңдап, оның қысқаша тарихын оқып, қойылған суреттерден табиғат сұлулығын көріп, сол демалыс орнына жету мүмкіндігімен таныса алады.

А.1.1 Өндеудің мақсаты мен қызметі

Қосымша пайдаланушы мен деректер қорының спецификасында жасалынды. Қосымша Android платформасында Android Studio бағдарламасының көмегімен Java программалау тілінде жазылды.

Барлық элементтер, яғни мәтін string-те жазылады. Қосымшанның клиенттік жағына Android Studio жауап берсе серверлік жағынан мәтін string-те, видеолар хостингтен алынуда.

Мәтін мен суреттерді жылжыту үшін ScrollView және HorizontalScrollView компоненттерін қолдындым. Экранның болашақ макетін жасауда бағдарламадан өзімізге қажет шаблонды таңдаймыз. Activity Name – қосымшаның басты классының атауы. Менде ол Мәдениет деп аталады. Layout Name – xml файлының аты, онда визуалды интерфейс элементтері сақталады.

А қосымшасының жалғасы

А.1.2 Қолдану саласы

А Қосымшасының жалғасы

Қолдану аймағы – шет елден келген қонақтар мен туристтерге арнайы жасалған қосымша.

А.1 Анықтамалар, терминдер және қысқартулар

UML (ағыл. Unified Modeling Language – бірыңғай модельдеу тілі) – бұл графикалық сипаттама тілі Бағдарламаның нысаны сипатын әзірлеу үшін пайдаланылады. UML-модельі болып аталынатын графикалық символдарды пайдаланып жүйенің абстрактілік модельін құру. UML бағдарламалау тілі болып табылып, UML модельдері негізінде кодты генерациялауы мүмкін.

Android операциялық жүйесі смартфондарға, планшеттерге, электронды оқулықтарға, нетбуктар мен смартбуктерге арналған жүйе. Дүние жүзіне ең танымал мобильді қосымшалардың бірі. Ол-планшет, смартфондар, нетбук, электронды кітаптарда, қол сағаттарда жұмыс жасайтын Linux ядросының көмегімен жұмыс істейді.

Google Play – мобильді қосымшалар дүкені Google компаниясы қосымшаларына арналған. Android операциялық жүйесінде әр түрлі мобильді қосымшалардың ақылы және де ақысыз бағдарламаларды жүктеуіне арналған. Мобильді қосымша әзірлеушілері кез-келген елдің өздерінің ақылы қосымшаларын сату арқылы табыс көзін табуға да болады.

А.2 Жалпы сипаттамасы

Соңғы жылы смартфондармен планшеттердің сатылу көрсеткіші бірнеше мәртеге өскен. Осы көрсеткіштер үнемі өсуде, қазіргі уақытта бұл көрсеткіш әлі өзгермеген. Мобильді қосымшаларының өзекті және маңызды болуы айқын.

Бағдарламалық камсыздандырулар мен мобильді қосымшаларды дайындаушыларға сұраныс өте көп.

Мобильді қосымша дегеніміз – смартфондар мен басқа да қосымша құралдарда жұмыс жасайтын программалық қамтама. Барлық мобильді

А қосымшасының жалғасы

қосымшалар интернет желісі арқылы интернет магазиндері арқылы жүктелініп алынады. Менің мобильді қосымшам Android платформасында жазылған қосымша. Жобамда Алматы қаласымен Алматы облысына кіретін танымал демалыс орындарымен таныс боласыздар. Қосымшаның ерекшелігі пайдаланушы маңызды мәлеметтерді нақты ақпарат құралдарынан ала-алады, демалыс орындарының тарихы, суреті, мекен-жай карта көмегімен көрсетіледі.

Және қосымшаны үш тілде көру мүмкіндігі бұл қосымшаның өзіндің зор көрсеткішін жүзеге асыратын қызығушылығын алып келетін бағдарлама болып табылады. Жасалу бойынша ерекшелігін табылығандықтан керек болады.

А.2.1 Пайдаланушылық интерфейстер

Қолданушыға арналған интерфейсте қосымшаны қосуға, онда ақпарат алуға, суретін көруге, қысқаша ақпарат алуға болатын пайдаланушыға ыңғайлы және қарапайым етіп жасалған интерфейс құрастырылған. Қосымшаны іске қосқанда, бас мәзір шығады. Мәтін бөлімінде үш бөлімшелер бар. Олар экранның мәтін классының атауының астында бір бағанда кезекпен орналасқан. Әр бөлімшеде өзіне тақырыптық саласына сай элементтер орналасқан.

А.2.2 Аппараттық интерфейстер

Құрылғыға қойылатын жалпы талаптар:

– Android платформасында жұмыс істейтін смартфон немесе Android 4.0.3 немесе одан жаңа нұсқалары.

А.2.3 Программалық интерфейстер

Серверлік программалы компоненттер:

- Android Studio;
- Java;
- БД Realm;

– Интернет хостинг.

А қосымшасының жалғасы

Клиенттік программалық компонент:

– Android операциялық жүйесі.

A.2.4 Коммуникациялық интерфейстер

Пайдаланушылардың телефондарына жүктеу үшін ғаламторға қатынау мүмкіндігі болуы керек. Мобильді құрылғыға жүктелгеннен кейін ғаламторсыз қолдана беруге болады.

A.2.5 Жады бойынша шектеулер

Қосымшаны пайдалануға арналған жалпы жады бойынша талаптар, телефонда қосымшаны жүктеуге жеткілікті жады болуы керек.

A.2.6 Коммуникациялық интерфейстер

Бұл мобильді қосымша тек Андроид операциялық жүйесіне орнатылған мобильдік құрылғыларда, смартфондарда, планшеттерде жұмыс істейді.

A.3 Өңдеуге талаптар

A.3.1 Функционалдық талаптар

Функционалдық талаптар негізінен бағдарламалық камтаманың атап айтқанда нені істейтінін анықтайды, қандай тапсырмаларды шешетінін анықтайды. Бұл мобильдік қосымшада тестілеуде, жұмысты қолданушы тұрғысынан істейміз. Тестілеу кезінде келесі амалдарды орындаймыз:

- қосымшаны іске қосу;
- барлық элементтің көрініп тұрғандарына көз жеткізу;
- мәзір бетіне өту;

– мәзірде бөлімді таңдау;

А қосымшасының жалғасы

– таңдалған бөлімге өту;

– мәзірдегі барлық бөлімді тексеру;

– бөлімдердің дұрыс жұмыс істеп тұрғандығына көз жеткізу.

Б Қосымшасы (міндетті)

Бағдарлама мәтіні

```
//AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android=http://schemas.android.com/apk/res/android
  xmlns:tools=http://schemas.android.com/tools
  package="com.example.tour">
  <uses-permission android:name="android.permission.INTERNET" />
  <application
    android:name=".TourApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true"
    tools:ignore="AllowBackup,GoogleAppIndexingWarning,Unused Attribute
">
  <activity
    android:name=".ui.regions.RegionsInfoActivity"
    android:screenOrientation="portrait"
    android:theme="@style/AppTheme.NoActionBar"
    android:windowSoftInputMode="adjustResize" />
  <activity
    android:name=".ui.category.InfoActivity"
    android:screenOrientation="portrait"
    android:theme="@style/AppTheme.NoActionBar"
    android:windowSoftInputMode="adjustResize" />
  <activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:theme="@style/AppTheme.Blue.NoActionBar" />
  <activity
    android:name=".SplashScreenActivity"
    android:theme="@style/AppTheme.NoActionBar">
  <intent-filter>
```

Б Қосымшасының жалғасы

```
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".LanguageActivity"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".WebViewActivity"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".ui.category.CategoryDetailActivity"
    android:theme="@style/AppTheme.Blue.NoActionBar" />
<activity
    android:name=".ui.regions.RegionsDetailActivity"
    android:theme="@style/AppTheme.Blue.NoActionBar" />
</application>
</manifest>
```

//

```
//MainActivity.java
package com.example.tour;import android.annotation.SuppressLint;import
android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import com.example.tour.ui.AboutFragment;
import com.example.tour.ui.SupportFragment;
```

Б Қосымшасының жалғасы

```
import com.example.tour.ui.TravellerFragment;
import com.example.tour.ui.category.CategoryFragment;
import com.example.tour.ui.regions.RegionsFragment;
import butterknife.ButterKnife;
public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    private Toolbar toolbar;
    public static Intent newIntent(Context packageContext) {
        return new Intent(packageContext, MainActivity.class);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.bind(this);
        toolbar = findViewById(R.id.toolbar);
        toolbar.setTitle("");
        setSupportActionBar(toolbar);
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navi
gation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        navigationView.setCheckedItem(R.id.nav_category);
        getSupportFragmentManager().beginTransaction().replace(R.id.content_frame, new CategoryFragment()).commit();
    }
    //Needed for locale change!
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(TourApplication.localeManager.setLocale(base));
    }
    @Override
    public void onBackPressed() {
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
```

Б Қосымшасының жалғасы

```
        drawer.closeDrawer(GravityCompat.START);
    } else {
        new AlertDialog.Builder(this)
            .setMessage(getString(R.string.alert_close_app))
            .setCancelable(false)
            .setPositiveButton(getString(R.string.alert_positive), new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    MainActivity.super.onBackPressed();
                }
            })
            .setNegativeButton(getString(R.string.alert_negative), null)
            .show();
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    //noinspection SimplifiableIfStatement
    // if (id == R.id.action_settings) {
    //     return true;
    // }
    return super.onOptionsItemSelected(item);
}
@SuppressWarnings("MissingPermission")
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    Fragment fragment = null;
    switch (item.getItemId()) {
```

Б Қосымшасының жалғасы

```
case R.id.nav_category:
    toolbar.setTitle("");
    fragment = new CategoryFragment();
    break;
case R.id.nav_regions:
    toolbar.setTitle("");
    fragment = new RegionsFragment();
    break;
case R.id.nav_traveller:
    toolbar.setTitle("");
    fragment = new TravellerFragment();
    break;
case R.id.nav_about:
    toolbar.setTitle("");
    fragment = new AboutFragment();
    break;
case R.id.nav_flight:
    Intent intFlight = new Intent(this, WebViewActivity.class);
    intFlight.putExtra("url", "http://www.aviasales.kz");
    startActivity(intFlight);
    break;
case R.id.nav_bus:
    Intent intBus = new Intent(this, WebViewActivity.class);
    intBus.putExtra("url", "http://xn--80aa0arnd.kz/booking");
    startActivity(intBus);
    break;
case R.id.nav_car:
    Intent intCar = new Intent(this, WebViewActivity.class);
    intCar.putExtra("url", "https://anytime.kz");
    startActivity(intCar);
    break;
case R.id.nav_hotel:
    Intent intHotel = new Intent(this, WebViewActivity.class);
    intHotel.putExtra("url", "http://www.booking.com");
    startActivity(intHotel);
    break;
case R.id.nav_lang:
    startActivity(new Intent(this, LanguageActivity.class));
    finish();
```

Б Қосымшасының жалғасы

```
        break;
    case R.id.nav_support:
        toolbar.setTitle("");
        fragment = new SupportFragment();
        break;
    case R.id.nav_sos:
        String phone = "112";
        Intent intent = new Intent(Intent.ACTION_DIAL, Uri.fromParts("tel", phone,
null));
        startActivity(intent);
        break;
    }
    if (fragment != null) {
        getSupportFragmentManager().beginTransaction().replace(R.id.content_frame,
fragment).commit();
    }
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
}

//
//TourApplication.java
package com.example.tour;
import android.app.Activity;
import android.app.Application;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.res.Configuration;
import android.os.Bundle;
import com.example.tour.helper.Key;
import com.example.tour.helper.LocaleManager;
public class TourApplication extends Application implements
Application.ActivityLifecycleCallbacks {
    private SharedPreferences prefs;
    public static LocaleManager localeManager;
    @Override
    public void onCreate() {
```


Б Қосымшасының жалғасы

```
    super.onCreate();
    registerActivityLifecycleCallbacks(this);
    prefs = getSharedPreferences(Key.SHARED_PREFS, MODE_PRIVATE);
}
@Override
public void onActivityCreated(Activity activity, Bundle bundle) {
}
@Override
public void onActivityStarted(Activity activity) {
}
@Override
public void onActivityResumed(Activity activity) {
}
@Override
public void onActivityPaused(Activity activity) {
}
@Override
public void onActivityStopped(Activity activity) {
}
@Override
public void onActivitySaveInstanceState(Activity activity, Bundle bundle) {
}
@Override
public void onActivityDestroyed(Activity activity) {
}
@Override
protected void attachBaseContext(Context base) {
    localeManager = new LocaleManager(base);
    super.attachBaseContext(localeManager.setLocale(base));
}
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    localeManager.setLocale(this);
}
}

\\
\\CategoryDetailActivity.java
```

Б Қосымшасының жалғасы

```
package com.example.tour.ui.category;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.ProgressBar;
import android.widget.TextView;
import com.example.tour.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;
import java.util.ArrayList;
import java.util.List;
import butterknife.BindView;
import butterknife.ButterKnife;
public class CategoryDetailActivity extends AppCompatActivity {
    @BindView(R.id.toolbar)
    Toolbar toolbar;
    @BindView(R.id.recyclerView)
    RecyclerView recyclerView;
    @BindView(R.id.category_name)
    TextView category_name;
    @BindView(R.id.progress)
    ProgressBar progress;
    private CategoryDetailAdapter categoryDetailAdapter;
    private String categoryId;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_category_detail);
        ButterKnife.bind(this);
```

Б Қосымшасының жалғасы

```
toolbar.setTitle("");
setSupportActionBar(toolbar);
if (getSupportActionBar() != null) {
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}
categoryId = getIntent().getStringExtra("category_id");
category_name.setText(getIntent().getExtras().getString("category_text"));
recyclerView.setHasFixedSize(true);
categoryDetailAdapter = new CategoryDetailAdapter(getApplicationContext(),
    new CategoryDetailAdapter.Callback() {
        @Override
        public void onItemClick(CategoryDetail categoryDetail) {
            Intent intent = new Intent(CategoryDetailActivity.this,
                InfoActivity.class);
//            intent.putExtra("category_id", categoryId);
            intent.putExtra("detail_id", categoryDetail.getDetail_id());
            intent.putExtra("detail_text", categoryDetail.getDetail_name());
            intent.putExtra("detail_image", categoryDetail.getDetail_image());
            intent.putExtra("detail_info", categoryDetail.getDetail_info());
            intent.putExtra("detail_address", categoryDetail.getDetail_address());
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        }
    });
recyclerView.setAdapter(categoryDetailAdapter);
/**
 * Simple GridLayoutManager that spans three columns
 */
GridLayoutManager manager = new GridLayoutManager(getApplicationContext(),
3, GridLayoutManager.VERTICAL, false);
recyclerView.setLayoutManager(manager);
progress.setVisibility(View.VISIBLE);
fetchCategoryDetails();
}
private void fetchCategoryDetails() {
    FirebaseFirestore.getInstance().collection("Data")
        .document("tour").collection("Category")
        .document(categoryId).collection("categoryDetail")
        .get()
    }
```

Б Қосымшасының жалғасы

```
.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
    @Override
    public void onComplete(@NonNull Task<QuerySnapshot> task) {
        if (task.isSuccessful()) {
            progress.setVisibility(View.GONE);
            List<CategoryDetail> categoryDetails =
mapDocumentToCategoryDetail(task.getResult()
                .getDocuments());
            categoryDetailAdapter.setDataset(categoryDetails);
        } else {
            Log.d("WWW", "Error getting documents: ", task.getException());
        }
    }
});
}
private List<CategoryDetail>
mapDocumentToCategoryDetail(List<DocumentSnapshot> documentSnapshots) {
    List<CategoryDetail> categoryDetails = new ArrayList<>();
    for (DocumentSnapshot d : documentSnapshots) {
        categoryDetails.add(new CategoryDetail(d.getId(),
            d.getString("detail_name"), d.getString("detail_image"),
d.getString("detail_info"), d.getString("detail_address")));
    }
    return categoryDetails;
}
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            finish();
            break;
    }
    return super.onOptionsItemSelected(item);
}
}
}
\\
\\RegionsDetailActivity.java
```

Б Қосымшасының жалғасы

```
package com.example.tour.ui.regions;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.bumptech.glide.Glide;
import com.bumptech.glide.request.RequestOptions;
import com.example.tour.R;
//import com.example.tour.ui.category.CategoryDetail;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;
import java.util.ArrayList;
import java.util.List;
import butterknife.BindView;
import butterknife.ButterKnife;
public class RegionsDetailActivity extends AppCompatActivity {
    @BindView(R.id.toolbar)
    Toolbar toolbar;
    @BindView(R.id.recyclerView)
    RecyclerView recyclerView;
    @BindView(R.id.regions_name)
    TextView regions_name;
    @BindView(R.id.regions_info)
    TextView regions_info;
    @BindView(R.id.regions_image)
    ImageView regions_image;
```

Б Қосымшасының жалғасы

```
@BindView(R.id.progress)
ProgressBar progress;
private RegionsDetailAdapter regionsDetailAdapter;
private String regionId;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_regions_detail);
    ButterKnife.bind(this);
    toolbar.setTitle("");
    setSupportActionBar(toolbar);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
    regionId = getIntent().getStringExtra("regions_id");
    regions_name.setText(getIntent().getExtras().getString("regions_text"));
    regions_info.setText(getIntent().getExtras().getString("regions_info"));
    Glide.with(this)
        .load(getIntent().getExtras().getString("regions_logo"))
        .apply(RequestOptions.placeholderOf(R.drawable.ic_loading)
            .error(R.drawable.ic_loading))
        .into(regions_image);
    recyclerView.setHasFixedSize(true);
    regionsDetailAdapter = new RegionsDetailAdapter(getApplicationContext(),
        new RegionsDetailAdapter.Callback() {
            @Override
            public void onItemClick(RegionDetail regionDetail) {
                Intent intent = new Intent(RegionsDetailActivity.this,
                    RegionsInfoActivity.class);
                intent.putExtra("detail_id", regionDetail.getDetail_id());
                intent.putExtra("detail_text", regionDetail.getDetail_name());
                intent.putExtra("detail_image", regionDetail.getDetail_image());
                intent.putExtra("detail_info", regionDetail.getDetail_info());
                intent.putExtra("detail_address", regionDetail.getDetail_address());
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
            }
        });
    recyclerView.setAdapter(regionsDetailAdapter);
}
```

Б Қосымшасының жалғасы

```
/**
 * Simple GridLayoutManager that spans three columns
 */
GridLayoutManager manager = new GridLayoutManager(getApplicationContext(),
3,
    GridLayoutManager.VERTICAL, false);
recyclerView.setLayoutManager(manager);
progress.setVisibility(View.VISIBLE);
fetchRegionDetails();
}
private void fetchRegionDetails() {
    FirebaseFirestore.getInstance().collection("Data")
        .document("tour").collection("Regions")
        .document(regionId).collection("regionDetail")
        .get()
        .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if (task.isSuccessful()) {
                    progress.setVisibility(View.GONE);
                    List<RegionDetail> regionDetails =
mapDocumentToRegionDetail(task.getResult()
                        .getDocuments());
                    regionsDetailAdapter.setDataset(regionDetails);
                    Log.d("UUU", regionDetails.toString());
                } else {
                    Log.d("WWW", "Error getting documents: ", task.getException());
                }
            }
        });
}
private List<RegionDetail> mapDocumentToRegionDetail(List<DocumentSnapshot>
documentSnapshots) {
    List<RegionDetail> regionDetails = new ArrayList<>();
    for (DocumentSnapshot d : documentSnapshots) {
        regionDetails.add(new RegionDetail(d.getId(),
            d.getString("detail_name"), d.getString("detail_image"),
d.getString("detail_info"), d.getString("detail_address")));
    }
}
```

Б Қосымшасының жалғасы

```
        return regionDetails;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                finish();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}

\\
\\BaseActivity.java
package com.example.tour;
import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import butterknife.ButterKnife;
import butterknife.Unbinder;
abstract class BaseActivity extends AppCompatActivity {
    private Unbinder unbinder;
    protected abstract int getLayoutResourceId();
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(getLayoutResourceId());
        unbinder = ButterKnife.bind(this);
    }
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(TourApplication.localeManager.setLocale(base));
    }
    @Override
    public void onDestroy() {
        unbinder.unbind();
        super.onDestroy();
    }
}
```


Б Қосымшасының жалғасы

```
}

\\
\\LanguageActivity.java
package com.example.tour;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import butterknife.ButterKnife;
import butterknife.OnClick;
public class LanguageActivity extends BaseActivity {
    public static Intent newIntent(Context packageContext) {
        return new Intent(packageContext, LanguageActivity.class);
    }
    @Override
    protected int getLayoutResourceId() {
        return R.layout.activity_language;
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//        setContentView(R.layout.activity_language);
        ButterKnife.bind(this);
    }
    @OnClick(R.id.btn_eng)
    void onClickEng() {
        TourApplication.localeManager.setNewLocale(this, 1);
        startActivity(MainActivity.newIntent(LanguageActivity.this));
        finish();
    }
    @OnClick(R.id.btn_rus)
    void onClickRus() {
        TourApplication.localeManager.setNewLocale(this, 0);
        startActivity(MainActivity.newIntent(LanguageActivity.this));
        finish();
    }
}
}
```

