ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Парман Жансая

Android платформасында кітаптарды сату интернет-дүкені

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

5В070400 – «Ақпараттану» мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
ПИ кафедра меңгерушісі
тех. ғыл. кандидаты,
ассистент-профессор
_____ Р. Юнусов
"17" мая 2019ж.

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: Android платформасында кітаптарды сату интернет-дүкені

5В060200 – «Ақпараттану» мамандығы

Орындаған Парман Жансая

Ғылыми жетекші
Сениор- лектор
_____ Р.С.Алғожаева
"17" 05 2019 ж.

Алматы 2019

Дипломдық жобаны орындау
КЕСТЕСІ

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

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

Бөлімдер атауы	**Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)**	**Қолтаңба қойылған мерзімі**	**Қолы**
Нормалық бақылаушы	Алибиева Ж. сениор-лектор	16.05.2019	
Бағдарламалық бөлім	Сман Н. ассистент		

Ғылыми жетекші _____ Р.С.Алғожаева

Тапсырманы орындауға қабылдап алған студент _____ Ж.Қ.Парман

Күні «29» 04 2019ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

5В060200 – «Ақпараттану»

БЕКІТЕМІН
ПИ кафедра меңгерушісі,
тех. ғыл. кандидаты,
Р. Юнусов
«17» мая 2019ж.

Дипломдық жоба орындауға
ТАПСЫРМА

Білім алушыға *Парман Жансая Құлжанқызы*
Тақырыбы: *«Android платформасында кітаптарды сату интернет-дүкені»*
Университет ректоры бұйрығының *№ 1841-б "14" наурыз 2019 ж.* шешімімен бекітілген.
Орындалған жобаның өткізу мерзімі *"20" мамыр 2019 ж.*
Дипломдық жобаның бастапқы мәліметтері: *Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма, жоба диаграммалары түрінде ақпаратты жинау, деректер қорына сақтау, тестілеу, тексеруге арналған программалық қамтамаларды жасау жүргізілген.*
Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:
а)тақырып таңдау және жобалау;
b) жобаның мақсат, міндеттерін талдау;
в)пайдаланушы интерфейсін жобалау және дамыту;
г) бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу;
Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):
презентацияның 20 слайдпен берілген құжат түрінде ұсынылған.
Ұсынылған негізгі әдебиеттер: *15 пайдаланылған әдебиеттер тізімінен*

АҢДАТПА

Дипломдық жобада қазіргі заманға сай кітап сату дүкеніне арналған Android мобильді қосымшаны құру қарастырылған. Мобильді қосымшаның артықшылығы тұтынушыларға әр түрлі жанрдағы кітаптарды ұсына отырып, тапсырыс жасауға тиімді қызмет көрсетуге мүмкіндік беруінде.

Дипломдық жұмыста кітап жеткізу қызметімен айналысатын жүйелерге шолу жасай отырып, мобильді қосымшаға қойылатын функционалдық және функционалды емес талаптарды талдауды қамтиды. Android операциялық жүйесінде мобильдік қосымшаларды әзірлеу технологияларын қолдана отырып, кітап сату дүкенінің халыққа тиімді мобильді қосымшасын құру іске асырылды.

АННОТАЦИЯ

Дипломный проект предусматривает создание мобильного приложения Android для современного магазина по продаже книг. Преимущество мобильного приложения в том, что оно позволяет клиентам эффективно обслуживать клиентов, предлагая книги разных жанров.

Дипломная работа включает обзор функциональных и нефункциональных требований к мобильным приложениям путем обзора систем, связанных с услугами доставки книг. Операционная система на базе Android разработала мобильное приложение, удобное для мобильных устройств, для магазина книг с использованием технологий разработки мобильных приложений.

ANNOTATION

The graduation project involves the creation of a mobile Android application for a modern store selling books. The advantage of the mobile application is that it allows customers to efficiently serve customers by offering books of different genres.

Thesis includes an overview of the functional and non-functional requirements for mobile applications by reviewing systems related to book delivery services. The Android-based operating system has developed a mobile application, convenient for mobile devices, for a book store using mobile application development technologies.

МАЗМҰНЫ

КІРІСПЕ

Бүгінгі таңда мобильді құрылғылар бізідің өмірімізге барынша енуде, тіпті сәнді зат болудан қалып, қажеттілігімізге айналды. Сол себептен де олар әртүрлі міндеттерді шешу үшін пайдаланылады. Мобильді құрылғылардың үлкен танымалдығы әзірлеушілер үшін еңбек нарығында жаңа нысанды құрды. Бірақ осыған қарамастан, мобильді қосымшаларды әзірлеу саласы жас және дамушы бағыт болып қала береді, әсіресе, web-әзірлеу немесе ойын әзірлеу салаларымен салыстырсақ.

Мобильді қосымшалар мен мобильді операциялық жүйелердің даму динамикасын 6-8 жыл бұрын смартфондарды пайдалана бастаған пайдаланушылар толық көлемде бағалай алады. Біраз уақыт бұрын келесі технологиялар бойынша белсенді патенттік талқылау жүрді: “slide to unlock”, “pinch to zoom”, “pull to refresh”. Ал, қазіргі таңда барлық құрылғылар мен операциялық жүйелерді өндірушілер осы технологияларды тең дәрежеде қолданады және бұл бірегей нәрсе емес. Айта кету керек, операциялық жүйелердің әр жаңартуларынан кейін олардың арасындағы принципті айырмашылықтар азая түсуде. Әрбір мобильді операциялық жүйе өз бағдарламалау тілінде жазылған. Нәтижесінде, егер тапсырыс беруші iOS және Android операциялық жүйелеріне арналған қосымшаларға тапсырыс берсе, онда әртүрлі бағдарламалық кодта жазылғанған, бірақ бірдей міндетті орындайтын қосымшалар әзірлеу қажет болады. Бірақ мобильді қосымшалардың кроссплатформлы әзірлемесі бар, алайда олар елеулі кемшіліктерге ие.

Қазіргі таңда көптеген компаниялар өз клиенттері үшін ғана емес, сонымен қатар қызметкерлері үшін де мобильді қосымшаларды әзірлей бастады, бұл жағдай осы саланың дамуын бағалаудың тағы бір тәсілі екенін атап өтуге болады. Мобильд технологияларды компанияның жұмыс процесіне енгізу келесі мақсаттарға қол жеткізуге мүмкіндік береді:

- еңбек өнімділігін арттыру;
- жұмыс процестеріне үлкен бақылауды қамтамасыз ету;
- ұйымның жалпы ақпараттануын арттыру.

Дипломдық жұмыстың мақсаты Android операциялық жүйесіне арналған кітап жеткізу қызметін жүзеге асыратын мобильді қосымшаны әзірлеу болып табылады.

Дипломдық жұмыстың мақсаты:

- ұқсас міндеттерді шешетін қолданыстағы жүйелерді зерттеу;
- мобильді қосымшаға қойылатын талаптарды талдау;
- Android қосымшаларын әзірлеу технологияларын талдау;
- мобильді қосымшаның архитектурасын жобалау;
- мобильді қосымшаны әзірлеу.

Бұл дипломдық жұмыс кіріспеден, төрт тараудан, қорытындыдан, пайдаланылған әдебиеттер тізімінен және қосымшадан тұрады. Жұмыс көлемі

35 беттен тұрады, пайдаланылған әдебиеттер тізімі – 10, қосымша көлемі – 5 бет.

Бірінші бөлім кітап жеткізу қызметімен айналысатын жүйелерді шолуға арналған.

Екінші бөлім мобильді қосымшаға қойылатын функционалдық және функционалды емес талаптарды талдауды қамтиды.

Үшінші бөлімде Android операциялық жүйесінде мобильдік қосымшаларды әзірлеу технологиялары қарастырылған.

Төртінші бөлімде мобильді қосымшаны іске асыру ұсынылған.

1 Қолданыстағы жүйелерді талдау

Қолданыстағы жүйелерді талдауға арналған ең төменгі талаптар:

– мобильді қосымша арқылы кітапқа тапсырыс жасау мүмкіндігі;

– тапсырыс статусын бақылау мүмкіндігі;

– тапсырыстар тізімін “Менің тапсырыстарым” бөлімі арқылы қарау мүмкіндігі.

Жүйенің қажетті функциялары:

– тапсырысқа комментарий қосу;

– төлем тәсілін таңдау;

– кітаптарды іздеу және жанр бойынша сұрыптау;

– кітаптарға баға беру, лүпіл басу және пікір жазу;

– қолданушыларға жеке деректерін өзгертулері үшін баптаулар панелін ұсыну;

– қолданушылар мекен-жайларларын қосымша жадысына сақтау.

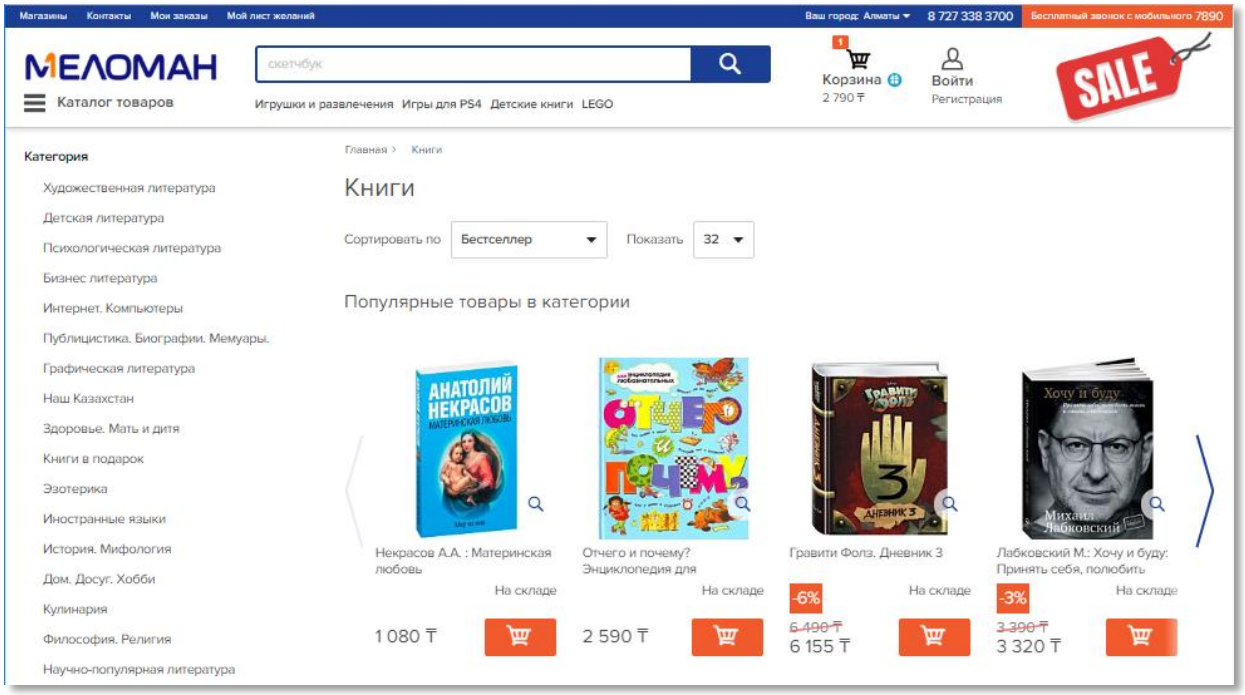
1.1 Ұқсас жүйелерлі шолу

Google PlayMarket мобильді қосымшалар дүкеніне жүргізген мониторинг нәтижесінде қолдану саласы бойынша ұқсас қосымшалар табылмады. Кітап сатуға байланысты жүйелер негізінен веб-сайт түрінде іске асырылған.

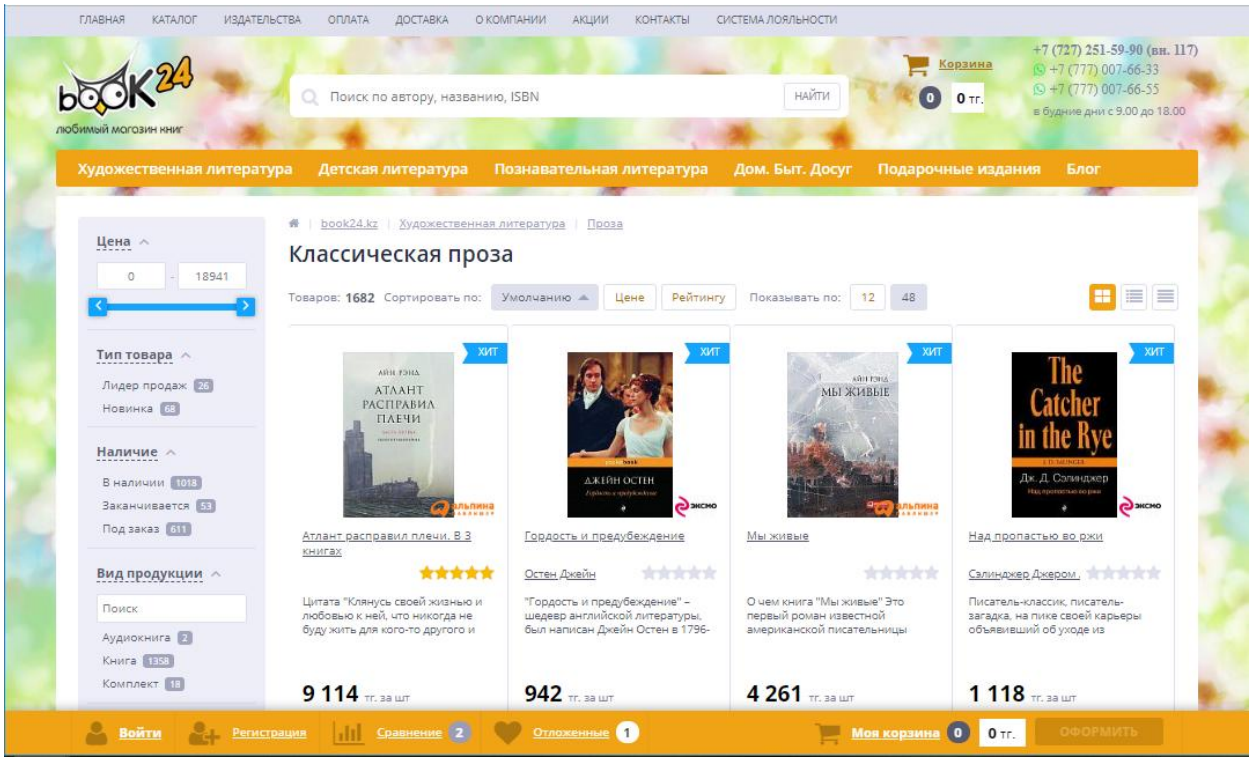
Менің ойымша, кітап сатуға және жеткізуге арналған ең жақсы веб-сайт – Меломан, ол 1.1-суретте бейнеленген. Барлық қарастырылған сайттардың ішінде ол ең жақсы функционалды ұсынады. Мысалы, кітаптардың үлкен ассортимені, кітіптар жайында егжей-тегжейлі ақпарат алу, қолданушыларға арналған бонустық бағдарлама, тәулігіне 24 сағат, аптасына 7 күн жұмыс істейтін анықтамалық колл-орталық. Сайтта тіркеу нысандары дұрыс құрастырылған, сонымен қатар кітаптарды сұрыптау қызметі жақсы форматта көрсетілген. Кітаптарға баға беруге және пікір жазуға болады. Бұл сайт өте сапалы және әдемі жасалған. Осы бағыттағы АТ инфрақұрылымын дамыту күн сайын танымалдылыққа ие болуда, ал АТ технологиялары өз кезегінде осындай жобаларды іске асыру үшін көптеген технологияларды ұсынады. Қарастырылған жүйе қолданушыларды мобильді қосымшамен қамтамасыз етпейді, бірақ қазіргі уақытта бұл кемшілік болып табылады. Бұл бағыттағы мобильді қосымша әзірлеудің маңызы зор, өйткені кітап сатуға және жеткізуге арналған лайықты шешімдер әлі жоқ.

1.2-суретте бейнеленген Book24 сайты алдыңғы сипатталған жүйе сияқты жақсы функционал мен дизайнға ие. Бұл сайттың басты ерекшелігіне кітаптар сипаттамаларын салыстыру қызметін жатқызуға болады. Мұнда бонустық бағдарлама қарастырылмаған.

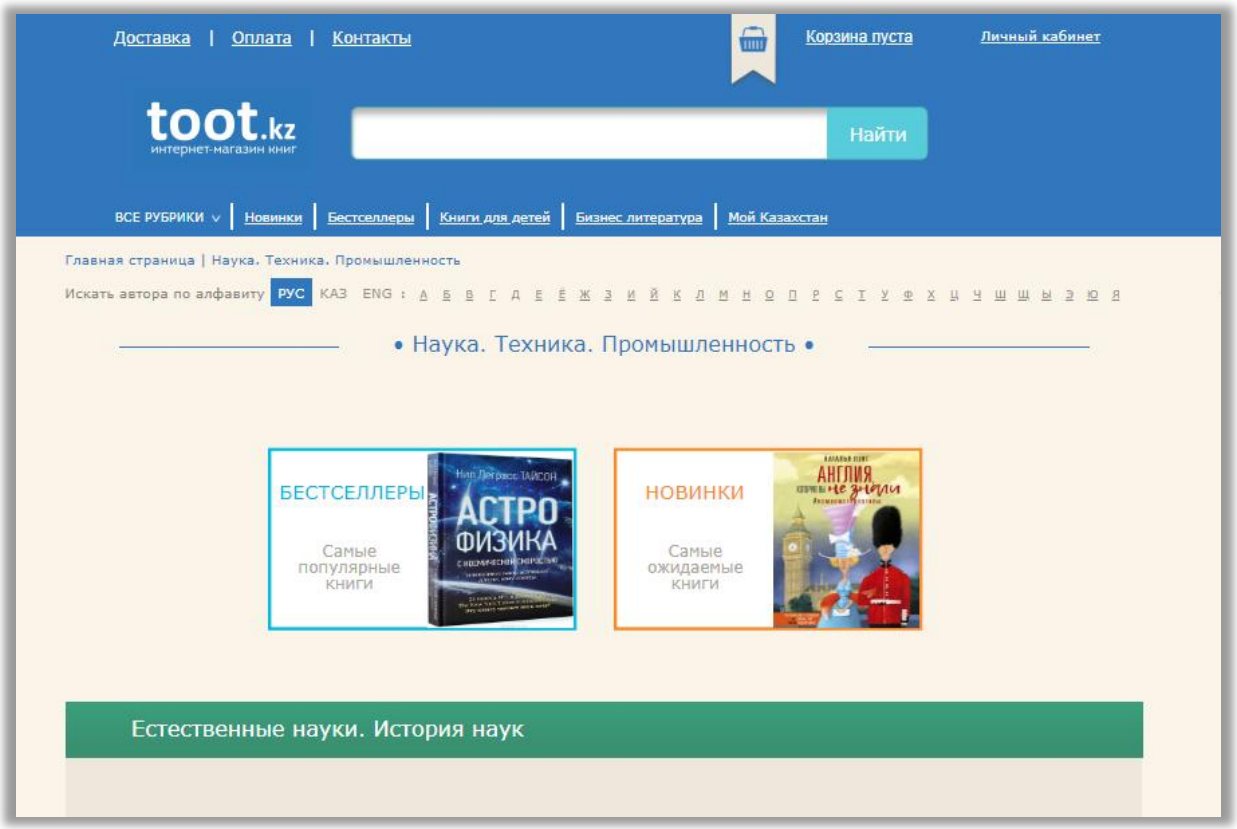
Toot.kz сайты 1.3-суретте бейнеленген, ол жоғарыдағы сипатталған сайттармен салыстырғанда әлдеқайда қарапайым. Бұл сайттың басты кемшілігі авторизацияның болмауында.



1.1-сурет – Меломан веб-сайты



1.2-сурет – Book24 веб-сайты



1.3-сурет – Toot.kz веб-сайты

2 Мобильді қосымшаға қойылатын талаптар

Мобильді қосымшаға қойылатын талаптарды талдау жоспарланған пайдалану сценарийі негізінде жүргізілді. Кітап тапсырыстарына жауапты компания қызметкерін “Кітап жеткізу” мобильді қосымшасы орнатылған ұялы телефонмен қамтамасыз ету күтілуде. Қызметкер өзінің телефон нөмірін және құпия сөзін қолдану арқылы авторизациядан өтеді, және соңғы келіп түскен тапсырыстарды қабылдау үшін қосымшаның тапсырыстарды басқару бетін синхрондайды немесе тапсырыстарды нақты уақыт режимінде қабылдайды. Тапсырыстарды ашып қарап, тапсырыс берушінің мекен-жайына кітаптарды жөнелтіп, тапсырысты тиісінше мәртебеге өзгертеді, және осы кезде тапсырыс берушінің мобильді құрылғысына мәртебенің өзгергені жайында хабарлама келеді. Мобильді қосымшаны қолдану барысында интернет желісі істен шыққан жағдайда, қолданушыға тиесілі ескерту жасалады. Сондай-ақ компания қызметкері мобильді қосымша арқылы деректер қорында орналасқан кітаптарды өзгерту, жою және жаңа кітаптарды қосу функцияларын нақты уақыт режимінде жүзеге асыра алады.

Қолданушылар, яғни тапсырыс берушілер кітап жеткізу қызметін қолдану үшін өздерінің мобильді құрылғыларына “Кітап жеткізу” мобильді қосымшасын орнатулары тиіс. Тапсырыс жасау үшін қажетті кітаптарды таңдап және міндетті түрде мекен-жайды толық толтырулары қажет. Қолданушылар кез-келген кітапқа өз бағасын беріп, пікірін қалдыра алады. Сонымен қатар, қолданушыларға сүйікті кітаптар және кітап іздеу функциялары қолжетімді.

2.1 Фунционалдық талаптар

Функционалдық талаптар жүйенің қалай іске асырылуы тиіс екендігі жайлы ақпаратты ұсынады, сондай-ақ ол жүйеде не іске асыру керектігін және жүйенің қандай мүмкіндіктерін іске асыратынын сипаттайды. Функционалдық талаптар бағдарлама шешуі тиіс пайдаланушылық есептер жиынтығын анықтайтын бағдарламалық қамтамасыз етуді сипаттайтын бизнес талаптарынан тұрады [1].

Сипатталған пайдалану сценарийі негізінде функционалдық талаптар құрылды. “Кітап жеткізу” мобильді қосымшасы үшін келесі функционалдық талаптар анықталды:

– қосымша сервер мен мобильді қосымша арасында деректерді синхрондау мүмкіндігін беруі керек;

– қосымша PUSH-хабарландыруларына қолдау көрсетуі қажет;

– қосымша пайдаланушының атқаратын рөліне қарай авторизациядан өтуге мүмкіндік беруі керек;

– қосымшада смартфондарға бейімделген интерфейс болуы керек;

– қосымша тапсырыстардың әр түрлі күйлерін дұрыс өңдеуі керек;

– қосымшада кітап іздеу, кітапқа баға беру, пікір жазу қызметтері қол жетімді болуы керек;

– қосымша төлем тәсілін таңдау мүмкіндігін және де төлем қолма-қол болған жағдайда қайтарым жасауға ақша соммасын енгізу функциясын қамтамасыз етуі қажет.

– қосымшада “Сүйікті кітаптар” қызметі болуы керек;

– қосымша қолданушының жеке деректерін өзгертуге мүмкіндік беруі керек.

Администраторға арналған мобильді қосымшаға қойылатын талаптар тізімі:

– қосымша сервер мен мобильді қосымша арасында деректерді синхрондау мүмкіндігін беруі керек;

– қосымша PUSH-хабарландыруларына қолдау көрсетуі қажет;

– қосымша пайдаланушының атқаратын рөліне қарай авторизациядан өтуге мүмкіндік беруі керек;

– қосымшада смартфондарға бейімделген интерфейс болуы керек;

– қосымша тапсырыстардың әр түрлі күйлерін дұрыс өңдеуі қажет;

– қосымша мобильді құрылғының камерасын қолдануға мүмкіндік беруі қажет;

– қосымша жаңа категория, жанр, кітап қосуға және оларды өзгертуге, жоюға мүмкіндік беруі қажет.

2.2 Функционалдық емес талаптар

Функционалдық емес талаптар мінез-құлықтың жекелеген сценарийлері емес, жалпы жүйенің жұмыс критерийлерін анықтайды. Функционалды емес талаптар өнімділік, техникалық қызмет көрсету ыңғайлылығы, кеңейтімділік, сенімділік, пайдаланудың орта факторлары сияқты жүйелік қасиеттерді анықтайды [1]. Мобильді қосымшаның қолданушылары үшін келесі функционалды емес талаптар қалыптастырылды:

– жүйе клиент – сервер – деректер базасының сервері архитектурасы негізінде жүзеге асырылуы тиіс;

– сервер Java бағдарламалау тілінің көмегімен жазылып, Firebase платформасында өрістелуі тиіс;

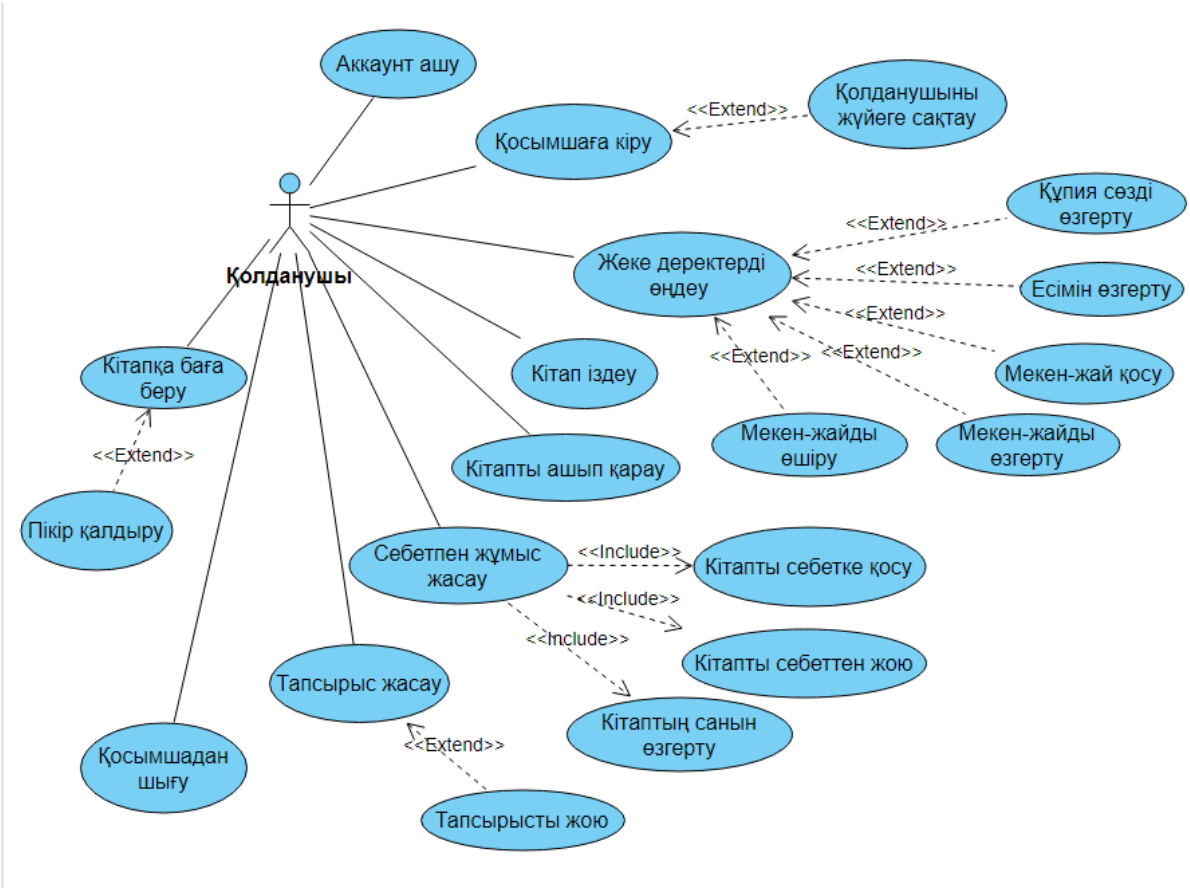
– сервер деректерді манипуляциялау үшін REST API ұсыну керек;

– мобильді қосымша Android Studio платформасының көмегімен Java бағдарламалау тілінде әзірленуі тиіс.

2.3 Прецеденттерді сипаттау

UML-дегі визуалды модельдеуді бастапқы жүйенің жалпы және абстрактілі тұжырымдамалық логикалық моделінен тиісті бағдарламалық жүйенің физикалық моделіне деңгейлеп түсудің процесі ретінде елестетуге болады. Бұл мақсаттарға жету үшін алдымен жүйенің функционалдық мақсатын сипаттайтын немесе, басқаша айтқанда, жүйе өзінің функционалдану процесінде не істейтінін сипаттайтын пайдалану нұсқалары диаграммасы құрылады. Пайдалану нұсқалары диаграммасы жүйенің бастапқы концептуалды көрінісі немесе жобалау және әзірлеу процесінде оның тұжырымдамалық үлгісі болып табылады.

Пәндік аймақтың мәндерін және талаптарды талдау нәтижесінде мобильді қосымша үшін пайдалану нұсқалары моделы қалыптастырылды. Ол 1.1-суретте көрсетілген. Осы модельге сәйкес қолданушының жүйемен жасайтын іс-әрекеттер тізімін айқын көруге болады.



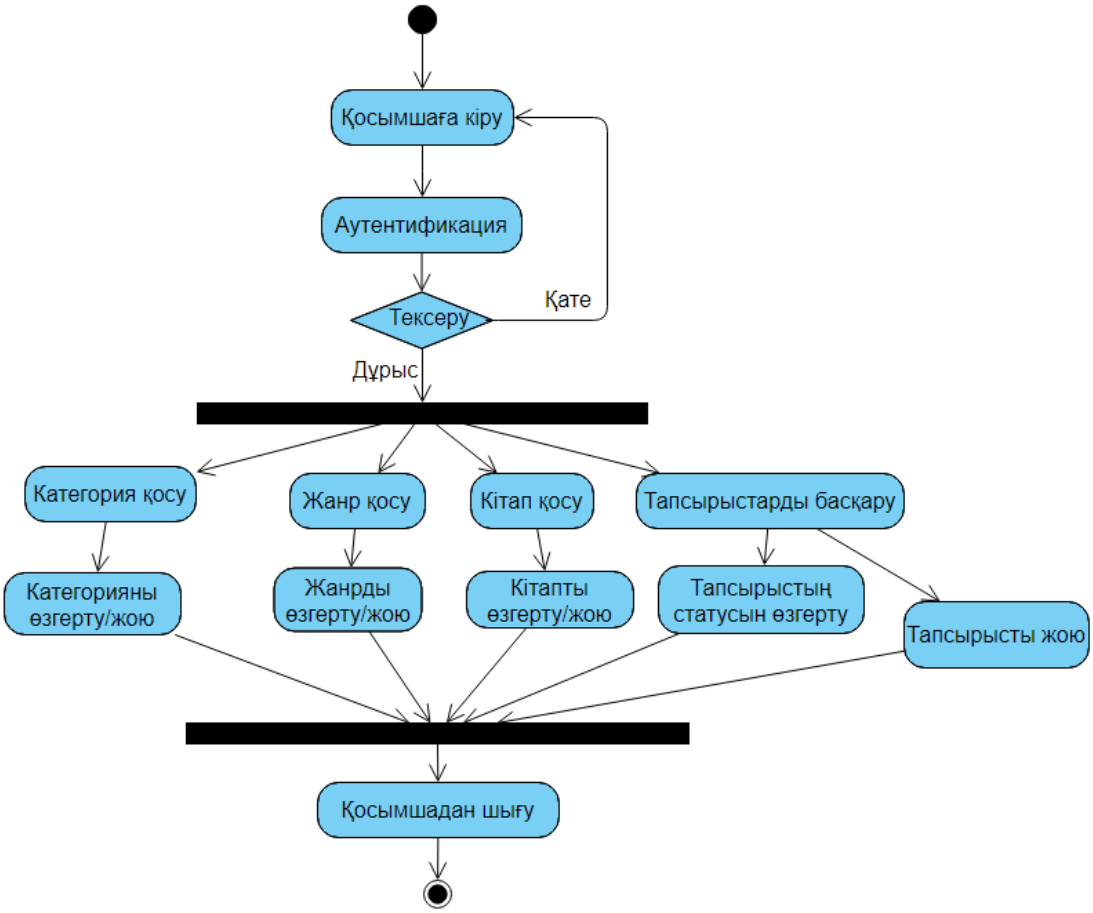
2.1-сурет – Пайдалану нұсқалары диаграммасы

Бұл диаграмманың мәні мыналардан тұрады: жобаланатын жүйе пайдалану нұсқалары арқылы жүйемен өзара әрекеттесетін актерлер немесе көптеген мәндер түрінде ұсынылады. Бұл ретте сырттан жүйемен өзара әрекеттесетін кез келген мән актер немесе әрекет етуші тұлға деп аталады. Бұл,

модельдік жүйеге әсер ету көзі ретінде әзірлеуші өзі анықтайтын, адам, техникалық құрылғы немесе кез келген басқа жүйе болуы мүмкін. Өз кезегінде, пайдалану нұсқасы актерге жүйемен ұсынылатын сервистерді сипаттау үшін қызмет етеді. Басқаша айтқанда, әр пайдалану нұсқасы актермен сұхбат кезінде жүйемен жасалатын әрекеттер жиынтығын анықтайды. Бұл ретте актерлердің жүйемен өзара іс-қимылы қалай іске асырылатыны туралы ештеңе айтылмайды.

2.4 Қызмет диаграммасы

Қызмет диаграммасы қандай да бір пайдалану нұсқасын орындау үшін қажетті іс-қимылдар реттілігін көрсетеді [2]. Яғни, бұл диаграмма прецеденттер диаграммасымен тығыз байланысты. Дәлірек айтқанда, пайдаланудың әрбір нұсқасы үшін белгілі бір прецедентті іске асыруға қажетті әрекеттер мен шешімдердің барысын көрсететін қызмет диаграммасын құруға болады .



2.2-сурет – Қызмет диаграммасы

Мәні жағынан қызмет диаграммалары алгоритмдердің қарапайым блок-схемаларына өте ұқсас, тек қана мәні бойынша ғана емес, сондай-ақ өзінің

мақсаты мен құрамы бойынша да. Жоғарыдағы 1.2-сурет администратордың мобильді қосымшадағы жасайтын іс-әрекеттерін айқындайтын қызмет диаграммасы көрсетілген.

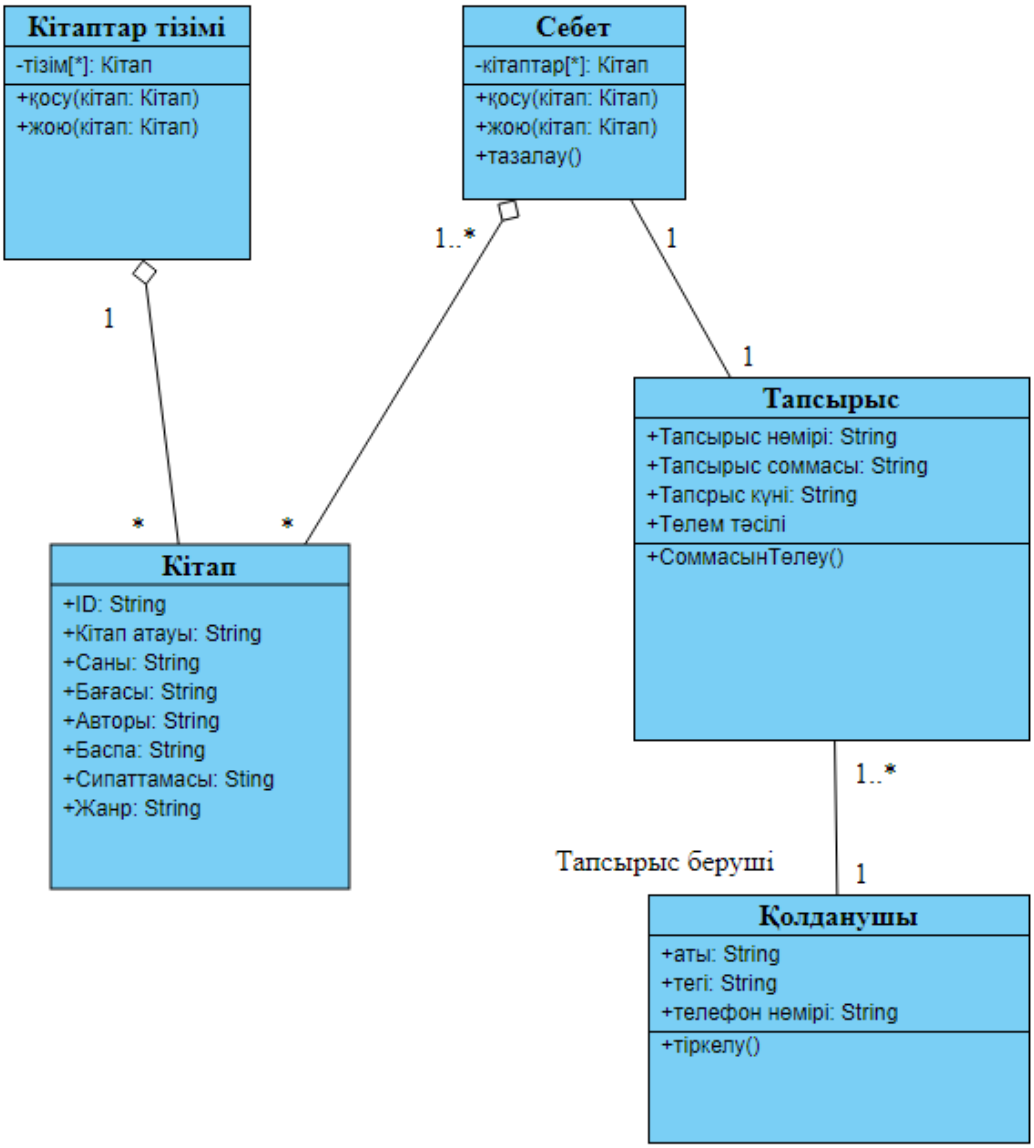
Администратор қосымшаға кіру үшін міндетті түрде жүйеге тіркелген болуы қажет. Жоғарыдағы диаграммада көрсетілгендей администратор қосымшаға кіргеннен кейін, келесі іс-әрекеттерді жүзеге асыра алады: категория қосу, категорияны өзгерті/жою, жанр қосу, жанрды өзгерту/жою, кітап қосу, кітапты өзгерту/жою, тапсырыстардың статусын өзгерту, тапсырыстарды жою.

2.5 Қосымшада кітаптарды сату жүйесіне үшін кластар диаграммасы

Android қосымшасында кітаптарды сату жүйесі үшін кластар диаграммасы 2.3-суретте көрсетілген. Жалпы айтқанда, кластардың осындай диаграммасы әртүрлі түрде құрылуы және ұсынылуы мүмкін, бұл модельді іске асырудың жекелеген аспектілеріне әзірлеушінің субъективті сипатын көрсетеді.

Бұл диаграмма бес класты қамтиды, бұл ретте қолданушы класы осы диаграмманың элементі болып табылады, бірақ ол пайдалану нұсқаларының диаграммасында актер ретінде бейнеленген болатын. Кластар операциясының құрамына және олардың көріністеріне келетін болсақ, бұл оларды таңдау осы модельді ықтимал бағдарламалық іске асыруға байланысты анықталады.

Кластардың бейнеленген диаграммасы қосымшада кітаптарды сату жүйесінің алдын ала статикалық үлгісі болып табылады және жобамен жұмыстың келесі кезеңдерінде нақтылануы мүмкін. Атап айтқанда, жекелеген кластардың операциялар құрамын немесе олардың көрінісін өзгерту, қосымша атрибуттарды енгізу немесе операциялар сигналдарын неғұрлым егжей-тегжейлі сипаттап жазылуы мүмкін.



2.3-сурет – Қосымшада кітаптарды сату жүйесі кластарының диаграммасы

3 Android қосымшаларын әзірлеу технологияларын талдау

3.1 Деректерді сақтау технологияларын талдау

Android амалдық жүйесі деректерді сақтауға арналған технологиялардың әр түрлі типтерін қолдануға мүмкіндік береді: қосымшалар арасындағы деректер алмасуға мүмкіндік беретін SQLite және Content Provider реляционды деректер қоры – кірістірілген шешімдер болып табылады. Сонымен қатар деректерді сақтау үшін NoSql деректер қоры принциптерін жүзеге асыратын басқа да технологиялар бар. Деректерді сақтау технологияларын салыстырмалы талдау 1-кестеде келтірілген.

1-кесте – Android амалдық жүйесінде деректерді сақтау тәсілдері

	OrmLite	StorIO	Content Provider	Firebase
Деректерді сақтау тәсілі	SQLite	SQLite/ Content	Content Provider	NoSql
Күрделі сұрауларды жазу мүмкіндігі	Бар	Бар	Жоқ	Жоқ
RxJava қолдау	Жоқ	Бар	Жоқ	Бар
Артықшы-лықтары	Кең функционалдылық, ORM деңгейінде қолдау қатынастары.	Қолданыстағы қарапайымды-лық	Деректердің қатаң типтелуі, асинхронды шақыруларға бағытталған	Жұмыс жылдамды-ғы, қолдануға оңай
Кемшілік-тері	Артық код жазу	Артық код жазу	Шектелген функционал	Шектелген функционал

Android қосымшалары үшін пайдаланылатын сервер – бұл Oracle SQL, Microsoft SQL Server және PHP файлдары арқылы сервермен байланысқан MySQL. Кейіннен Android қосымшалары үшін деректерді сақтауға JSON форматын пайдаланатын Firebase платформасы пайда болды. Басқа серверлер деректерді сақтау үшін кесте форматын пайдаланады. Firebase NoSQL құрылымына негізделген. Firebase сияқты бұлтты серверлер өте аз. Мұндай серверлерге келесілерді жатқызуға болады: AWS Mobile Hub, CloudKit, Parse. AWS Mobile Hub – бұл мобильді қосымшаларды құруға, тестіден өткізуге және бақылауға арналған біріктірілген консоль [3]. CloudKit – бұл Apple компаниясының платформасы, ол деректерді және ресурстарды сақтауға

көмектеседі, бірақ тек iOS операциялық жүйесінің қосымшаларына арналған [4]. Parse – Facebook компаниясына тиесілі, ашық бастапқы коды бар сервер. Бұл сервер қазіргі таңда қол жетімсіз [5]. Мен, барлық қол жетімді серверлік платформаларды және деректер қорын саралай келе өз дипломдық жобама Firebase платформасын қолдануды жөн көрдім.

Firebase – 2011 жылы Эндрю Ли және Джеймс Трамплин негізін қалаған және 2014 жылы Google компаниясының иелігіне өткен, бұлтты қызметтерді жеткізуші американдық компания. Негізгі қызметі – бағдарлама әзірлеушілеріне деректерді бірнеше клиенттер арасында сақтауға және синхрондауға мүмкіндік беретін NoSQL класындағы бұлтты ДББЖ (Деректер Базаларын Басқару Жүйелері) . Firebase платформасында Android Және iOS операциялық жүйелеріне арналған қосымшалармен біріктіру ерекшеліктері қолдау тапты, JavaScript, Java, Objective-C және Node.js қосымшаларына арналған API жүзеге асырылды, сондай-ақ, JavaScript-фреймворктардың қатарынан REST стиліндегі деректер базасымен тікелей жұмыс жасауға мүмкіндік бар. Деректерді шифрлау үшін API қарастырылғын [6].

Firebase деректер қорының артықшылықтары:

– JSON деректер үшін нақты уақытта синхрондау. Firebase Realtime деректер қоры – қолданушылар арасында деректерді нақты уақыт режимінде сақтауға және синхрондауға мүмкіндік беретін NoSQL бұлттық деректер қоры болып табылады.

– құрылғымен бірге жұмыс істеу. Деректерді нақты уақытта синхрондау қолданушыларға кез келген веб немесе мобильді құрылғыдан өз деректеріне қатынуға мүмкіндік береді, бұл өз кезегінде қолданушыларға бір-бірімен ынтымақтасуға көмектеседі.

– қосымшаларды сервердің көмегінсіз жасау. Realtime деректер қоры мобильді және веб SDK арқылы жеткізіледі, сондықтан серверлердің қажеттілігінсіз қосымшаларды жасай аласыз. Сондай-ақ біз Cloud Function for Firebase қызметін пайдалану арқылы деректер қоры бастамасымен оқиғаларға жауап беретін бэкэнд кодын орындай аламыз.

– дербес пайдалану үшін оңтайландырылған. Қолданушылар дербес режимге ауысқанда, Realtime SDK деректер қоры қызмет көрсету және өзгерістерді сақтау үшін құрылғыдағы жергілікті кэшті пайдаланады. Құрылғы желіге қосылған кезде, жергілікті деректер автоматты түрде синхрондалады.

– пайдаланушы қауіпсіздігі. Әзірлеушілер үшін қарапайым және интуитивті аутентификацияны қамтамасыз ету үшін Realtime деректер қоры Firebase Authentication-мен біріктірілген.

Firebase деректер қоры келесі кемшіліктерге ие:

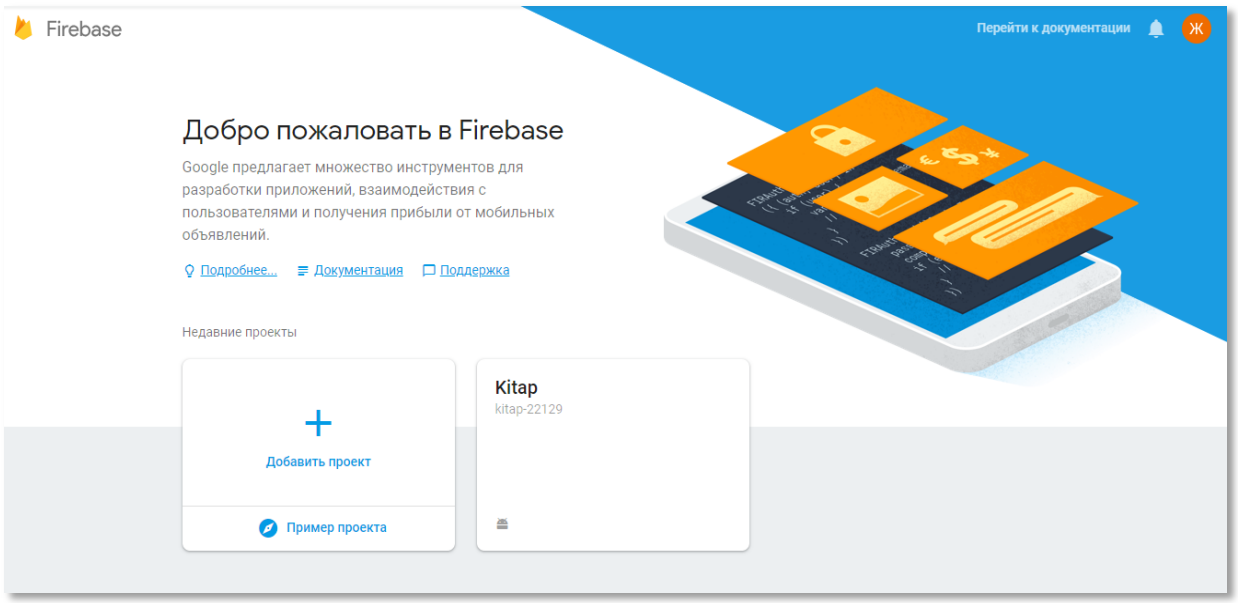
– оның қолданылу саласы NoSQL шешімдерінен әлдеқайда аз.

– Firebase деректерді алу кезінде және қажет болған жағдайда бір уақытта деректерді бірнеше орындарға жазу кезінде өзіндік шектеулерге ие.

– кейбір деректер құрылымы Firebase-та жұмыс жасауға ыңғайсыз.

Мобильді қосымшаға Firebase деректер қорын қосу алгоритмі Firebase консолінде және ашық Android жобасында бірқатар міндеттерден тұрады:

– Firebase жобасын құру. Android-қосымшасына Firebase-ты ендірмес бұрын, Firebase жобасын құрып жобасы 3.1-сурет көрсетілген.



3.1-сурет – Firebase консоліндегі құрылған жоба

– мобильді қосымшаны Firebase жобасына тіркеу.

– Firebase конфигурациялық файлын қосу. Google-services.json файлын жүктеп, жобадағы app пакетіне жылжытамыз. Firebase өнімдерін қосымшаға қосу үшін google-services плагинін Gradle файлдарына қосамыз.

– жоба деңгейіндегі Gradle файлының көрінісі 3.2-суретте көрсетілген.

– ал, қосымша деңгейіндегі Gradle файлының көрінісі 3.3-суретте көрсетілген.

– қосымшаға Firebase SDK қосу. Модульдің Gradle файлына (бағдарлама деңгейінде, әдетте app / build.gradle) негізгі Firebase SDK үшін тиісті тәуелділікті қосу қажет. Осы суретте көрсетілген тәуелділіктердің әрбіріне жеке-жеке тоқталып өтейін.

FirebaseUI – бұл Android платформасы үшін ашық бастапқы коды бар кітапхана, ол пайдаланушы интерфейсінің жалпы элементтерін Firebase API-интерфейстеріне жылдам қосуға мүмкіндік береді. FirebaseUI нақты уақыттағы Firebase деректер қоры, Бұлттық Firestore, Firebase Auth және Бұлтты Сақтау үшін жеке модульдерге ие [7].

Firebase Аналитика бүгінгі күні қол жетімді мобильді қосымшалар талдаушыларының ең танымал шешімдерінің бірі болып табылады. Оны пайдалана отырып, қосымша қолданушылары туралы нақты ақпарат алуға болады. Жалпы, Firebase Аналитика екі негізгі ақпарат түрін тіркейді:

Оқиғалар – қосымшада не болып жатқанын, мысалы пайдаланушы әрекеттері, жүйелік оқиғалар немесе қателер;

Пайдаланушы қасиеттері – пайдаланушы базасының сегменттерін сипаттау үшін анықтайтын атрибуттар, мысалы, тілдік артықшылық немесе географиялық орын.

Аналитика кейбір оқиғалар мен пайдаланушы қасиеттерін автоматты түрде тіркейді: оларды қосу үшін код жазудың қажеті жоқ.

```
buildscript {
  // ...
  dependencies {
    // ...
    // Келесі жолды қосу:
    classpath 'com.google.gms:google-services:4.0.1'  // Google
Қызметтер плагині
  }
}

allprojects {
  // ...
  repositories {
    // Төмендегі жол міндетті түрде болу қажет
    google()  // Google-дың Maven репозиторийі
    // ...
  }
}
```

3.2-сурет – Жоба деңгейіндегі Gradle файлы

```
apply plugin: 'com.android.application'

android {
  // ...
}

//Файлдың соңына келесі жолды қосыңыз:
apply plugin: 'com.google.gms.google-services'  // Google Play
қызметтері Gradle плагин
```

3.3-сурет – қосымша деңгейіндегі Gradle файлы

Firebase Бұлттық Хабарландыру (Firebase Cloud Messaging немесе FCM) iOS, Android платформаларында және Интернетте хабарлар мен хабарламаларды тегін жеткізуге және алуға мүмкіндік беретін сервер мен құрылғылар арасында сенімді және тиімді байланысты қамтамасыз етеді. FCM SDK барлық қажетті рұқсаттарды, сондай-ақ қабылдаушының қажетті функционалдығын автоматты түрде қосады.

```
dependencies {
 // Google аналитика
 implementation 'com.google.firebase:firebase-core:16.0.1'

 // Нақты уақыттағы Firebase деректер қоры
 implementation 'com.google.firebase:firebase-database:16.0.1'

 // Нақты уақыттағы Firebase деректер қоры үшін FirebaseUI
 implementation 'com.firebaseui:firebase-ui-database:1.2.0'

 // Firebase бұлттық хабарлама
 implementation 'com.google.firebase:firebase-messaging:17.0.0'
}
```

3.4-сурет – Тәуелділікті қосу

Firebase платформасының ең қуатты функцияларының бірі – бұл нақты уақыт режиміндегі деректер базасы: онда орындалатын барлық жазба операциялары оны қадағалайтын барлық клиенттер үшін бірден қол жетімді. Бұл деректер қорының көмегімен пайдаланушыларға бір құрылғыдан басқасына кедергісіз ауысуға, сондай-ақ басқа пайдаланушылармен бірден әрекеттесуге мүмкіндік беретін қолданбаларды жасауға болады.

Деректер JSON форматында сақталады және әрбір қосылған клиентпен нақты уақыт режимінде синхрондалады. Бұл дегеніміз, деректер қорындағы мәліметтер кілттік сөз және мән жұптары ретінде көрініс табады, мұндағы кілттік сөздер әрқашан жолдар, ал мәндер – примитивтер, массивтер немесе басқа JSON объектілері болып табылады. Нақты уақыт режиміндегі деректер қорына мәліметтерді жазар алдында FirebaseDatabase класының getInstance() әдісі арқылы алдын ала деректер қорына сілтеме алу керек. Содан кейін getReference() әдісі арқылы деректер қорының ішіндегі кез келген жолға сілтеме алуға болады. Алынған жолда иерархиялық тәртіпте орналасқан бір немесе бірнеше кілттер болуы мүмкін. Егер жолда DatabaseReference бар болса, setValue() әдісін пайдаланып оның мәнін орнатуға болады. 3.5-суретте setValue() әдісі арқылы деректер қорына жаңа кітап қосу коды көрсетілген.

Дерекқордан мәліметтерді нақты уақыт режимінде оқу үшін асинхронды бақылаушыны байланысқан кілтке немесе жолға қосу керек. Дәлірек айтқанда, addValueEventListener() әдісін пайдалана отырып, ValueEventListener объектісін DatabaseReference объектісіне қосу керек. ValueEventListener интерфейсінің onDataChange() әдісі DataSnapshot объектісіне қатынауға мүмкіндік береді, ал getValue() әдісі кілттің соңғы мәнін алуға пайдаланылады. Мәселен, деректер қорындағы барлық кітіптар тізімін алу жолы 3.6-суретте көрсетілген.

```
// Деректер қорына жаңа кітап қосу:
  FirebaseDatabase database;
  DatabaseReference kitap;
  database = FirebaseDatabase.getInstance();
  kitap = database.getReference("Kitap");
  kosu.setOnClickListener(new View.OnClickListener() {
                  @Override
                  public void onClick(View v) {
                      if (zhanaKitap != null)
                      {
                          kitap.push().setValue(zhanaKitap);
                          Snackbar.make(rootLayout, "Жаңа кітап:
"+zhanaKitap.getAty()+" қосылды!", Snackbar.LENGTH_SHORT)
                                  .show();
                          finish();
                      }

                  }
              });
```

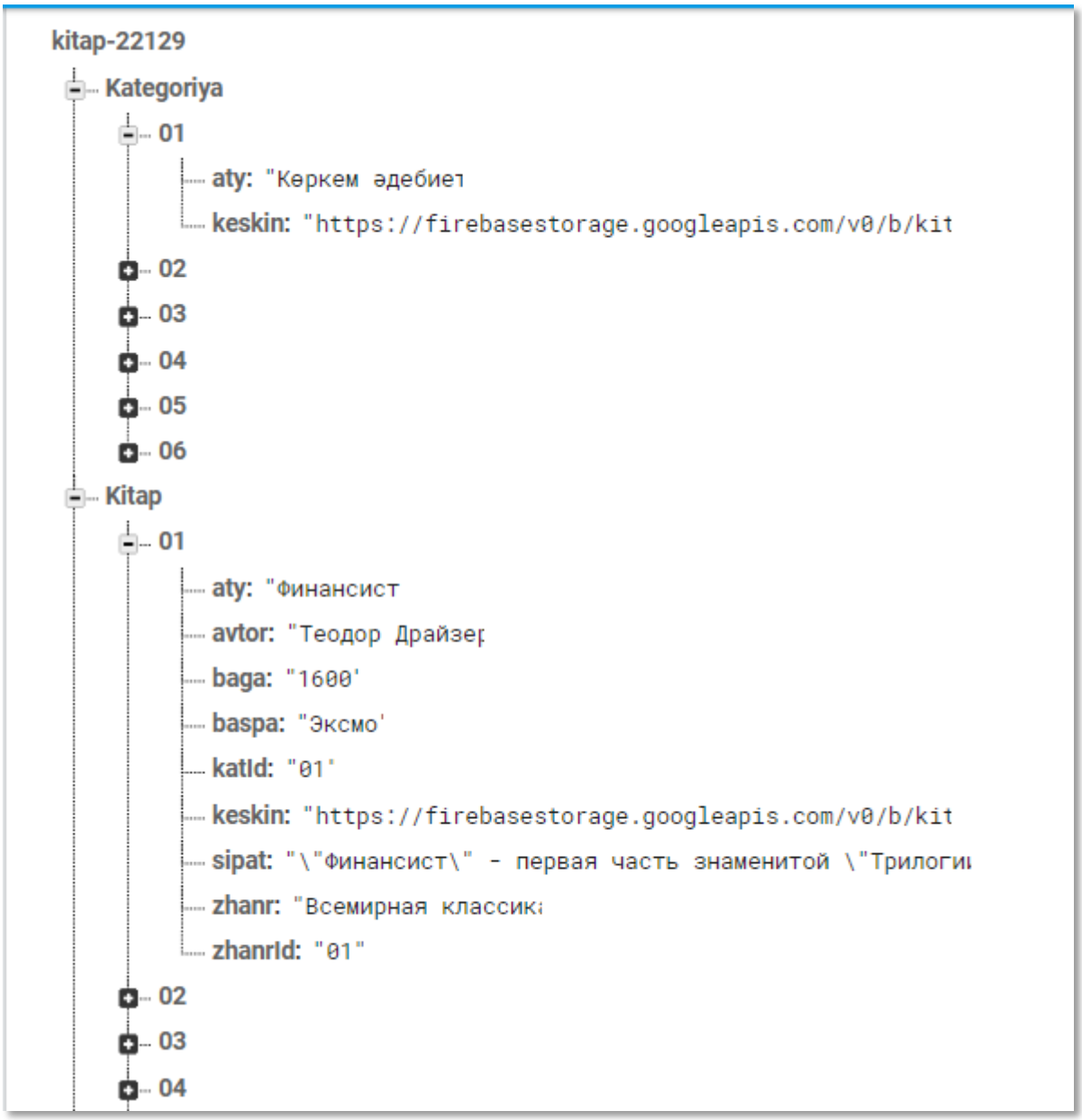
3.5-сурет – Деректер қорына жаңа кітап қосу

```
// Деректер қорындағы кітаптарды оқу
private void loadKitap(String katId) {
        adapterKitap = new FirebaseRecyclerAdapter<Kitap,
KitapViewHolder>(
                Kitap.class,
                R.layout.kitap_item,
                KitapViewHolder.class,
                kitap.orderByChild("katId").equalTo(katId)
        ) {

            . . .

   Query query=ratingTable.orderByChild("kitapId").equalTo
     (adapterKitap.getRef(position).getKey());
   query.addListenerForSingleValueEvent(new ValueEventListener() {

        @Override
   public void onDataChange(@NonNull DataSnapshot dataSnapshot){
   for (DataSnapshot postSnapshot:dataSnapshot.getChildren()) {
   Rating item = postSnapshot.getValue(Rating.class);
   }
        @Override
    public void onCancelled(@NonNull DatabaseError databaseError){

                    }
                });
```

3.6-сурет – Деректер қорынан кітаптар тізімін алу

IOS, Android және JavaScript үшін Firebase SDK көмегімен кроссплатформлы қосымшалар жасаған кезде, қосымшаның барлық қолданушылары нақты уақытта деректер қорының бір экземплярын бірлесіп пайдаланады және автоматты түрде ең жаңа деректер жаңартуларын алады. Нақты уақыттағы Firebase деректер қоры HTTP-сұраулардың орнына деректерді синхрондауды пайдаланады – әр кез деректер өзгерген сайын, кез келген жалғанған құрылғы бұл жаңартуды миллисекунд ішінде алады.

Firebase консолінің “Database” бөліміне өтіп, дерекқордың ішіндегі барлық деректерді нақты уақыт режиміндегі көріністі 3.7-сурет көре аламыз.



3.7-сурет – Нақты уақыттағы Firebase деректер қоры

3.2 Сервермен өзара әрекеттесуге арналған тәсілдерді талдау

Мобильді қосымшаға қойылатын талаптардың бірі сервермен синхрондау болып табылады. Бүгінігі таңда сервермен деректер алмасуға мүмкіндік беретін түрлі технологиялар бар. Синхрондау үдерісін HTTP сұраулары немесе socket байланыстары арқылы жасауға болады. Бірінші тәсіл web-әзірлеу саласында кең қолданысқа ие. Екіншісі дербес компьютерлерге арналған қосымшаларды әзірлеуде жиі қолданылады, және серверге қосылу қажет болған кезде әрдайым деректерді қабылдауды күтеді. Мұндай бағдарламалардың мысалы ретінде Skype, Icq сияқты хабар алмасу қосымшаларын жатқызуға болады [8].

Бағдарлама мен сервер арасындағы деректер алмасуды әртүрлі форматтарда деректерді жіберу арқылы жасауға болады: XML немесе JSON. Біздің пайдалану сценарийімізде JSON XML-ді, пайдаланудың қарапайымдылығына және сол ақпаратқа ие деректер пакетінің аз мөлшеріне байланысты, толығымен ауыстырды.

JSON-ның негізгі артықшылықтарына төмендегілерді жатқызуға болады:

– код оқылуы;

– кеңейту жеңілдігі.

Бүгінгі күні барлық мобильдік қосымшалар сервермен деректерді REST API көмегімен алмасады, оның негізгі артықшылықтары:

– трафикті үнемдеу – біз тек сұралған деректерді ғана аламыз;

– жүзеге асырудағы қарапайымдылық.

Кемшілігіне әр сұраныстың жеке түрде орындалатындығын жатқызуға болады.

REST API интерфейсіне балама ретінде деректердің барлығы бірден жүктелетін алмасу тәсілін ерекшелеуге болады. Бұл тәсілдің артықшылығы – іске асырудың қарапайымдылығында, яғни біз барлық деректерді бірден аламыз, әрбір сұрауды жеке-жеке өңдеу қажет емес.

Кемшіліктеріне келесілерді жатқызуға болады:

– тұтынылатын трафиктің үлкен көлемі;

– сұраныстың ұзақ уақыт орындалуы, әсіресе интернетке қосылу баяу болған кезде.

3.3 Желілік сұраныстарды іске асыратын технологияларды талдау

Android операциялық жүйесінде HTTP деректер алмасуды іске асыру үшін екі негізгі шешім бар: Retrofit және Java тілінің стандартты механизмдері. Retrofit-тің артықшылықтарына қолданудағы ыңғайлылықты жатқызуға болады, яғни сервермен деректер алмасу JSON форматында орын алады [9].

GSON технологиясы арқасында серверден келген жауап Java класы ретінде ұсынылуы мүмкін, осылайша серверден JSON жауабын қолмен талдаудың қажеті болмайды. Кемшіліктерінің ішінен жіберуге және қабылдауға арналған сұрау тақырыптарымен жұмыс жасау барысындағы ыңғайсыздықты ерекшелеуге болады. Стандартты Java кластары арқылы желілік деректерді алмасуды жүзеге асыру тікелей қарама-қарсы артықшылықтар мен кемшіліктерге ие. Артықшылықтары: сұрау тақырыптарымен жұмас істеу кезіндегі икемділік. Кемшіліктері: серверден келген жауапты “қолмен” талдау, бірақ GSON техгологиясын автоматты емес режимде қолдану бұл мәселені ішінара шешеді. Сондай-ақ, кемшіліктеріне артық бағдарламалық код жазу қажеттілігін жатқызуға болады.

REST API арқылы қосымшаның сервермен өзара әрекеттесуін қамтамасыз ету үшін Retrofit кітапханасын қолдану коды төменде көрсетілген.

```
buildscript {
  package kz.zhansaya.kitap.Remote;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RetrofitClient {
    private static Retrofit retrofit=null;

    public static Retrofit getClient(String baseURL)
    {
        if (retrofit == null)
        {
            retrofit = new Retrofit.Builder()
                    .baseUrl(baseURL)

.addConverterFactory(GsonConverterFactory.create())
                    .build();
        }
        return retrofit;
    }
}
```

3.8-сурет – Retrofit кітапханасын қолдану коды

4 Қосымшаны жүзеге асыру

4.1 Клиент қосымшасын жүзеге асыру

Клиенттік қосымшаны жүзеге асыру үшін Android Studio интеграцияланған әзірлеу ортасы (IDE) қолданылды. Бұл IDE бірқатар артықшылықтарға байланысты таңдалды:

– IDE әзірлеушісі Android платформасын шығарушы Google компаниясы. Бұл жаңа Android нұсқасы үшін қолайлы мүмкіндіктер интеграциясын береді;

– IDE-де API-дың барлық нұсқаларымен жұмыс істеуге мүмкіндік беретін Android SDK орнатылған, сонымен қатар барлық қажетті параметрлер автоматты түрде орнатылады;

– интерфейс конструкторы. Бәрі визуалданған, экран көрсетілуін кез-келген құрылғы мысалында көруге болады, сондай-ақ таңдалған құрылғының барлық сипаттамалары сақталады. Визуализация операциялық жүйенің нақты нұсқасында көрінеді;

– жоба құрылымы ұсынылады;

– ыңғайлы дизайн;

– қателерді бақылауға ыңғайлы.

4.2 Деректерді сақтау

Қосымша беттері арасында деректерді беру үшін putExtra() әдісі пайдаланылды, ол шын мәнінде кілт-мән жұбы түрінде ұсынылатын деректердің сөздігі болып табылады, сонымен қатар оны шақыру Intent класы арқылы жүзеге асырылады. PutExtra() әдісі арқылы сандық, жол және логикалық мәндер жіберіледі. Қосымшаны іске асыру кезінде белгілі бір шарттарға байланысты баптау файлдары немесе Intent класының объектілірі пайдаланылды [10].

Клиенттік қосымшада қолданушыларды жүйеге сақтау үшін, яғни қосымшаға автоматты түрде кіру қызметін енгізу үшін Paper кітапханасын пайдаландым. Paper кітапханасы мәліметтерді оңай және тез сақтауға мүмкіндік береді. Бұл кітапхананы қолдану үшін Gradle файлына келесі тәуелділікті енгізу қажет: implementation ‘io.paperdb: paperdb:2.6’. Application.onCreate() әдісінде бір рет инициализациялануы керек: Paper.init() [11].

Мобильді қосымшада кітаптарды себетке, сүйікті кітаптар бөліміне жинау және мекен-жайды сақтау қызметтерін жүзеге асыру үшін SQLite деректер қорын қолдандым. Ол үшін SQL тілінің синтаксисін пайдала отырып,

“KitapDB” деректер қорын құрдым. Менің құрған деректер қорым қосымша каталогындағы келесі жолда орналасқан: DATA/data/Kitap/databases/KitapDB.

SQLite – Android платформасына кіріктірілген кең таралған деректер қорын басқару жүйелерінің бірі. Дерекқорлармен жұмыс істеудің негізгі функцияларын android.database пакеті қамтамасыз етеді. Ал, SQLite деректер қорымен жұмыс істеуге мүмкіндік беретін кластар жиынтығы android.database.sqlite пакетінде орналасқан [12]. Осы кластардың әрқайсысына жеке тоқталып өтейін:

– android.database.sqlite.SQLiteDatabase класы деректер қорына сұрау- лар мен әртүрлі манипуляциялар жасауға мүмкіндік береді;

– android.database.sqlite.SQLiteCursor класы сұрауды қамтамасыз етеді және осы сұрауға сәйкес келетін жолдар жиынын қайтаруға мүмкіндік береді;

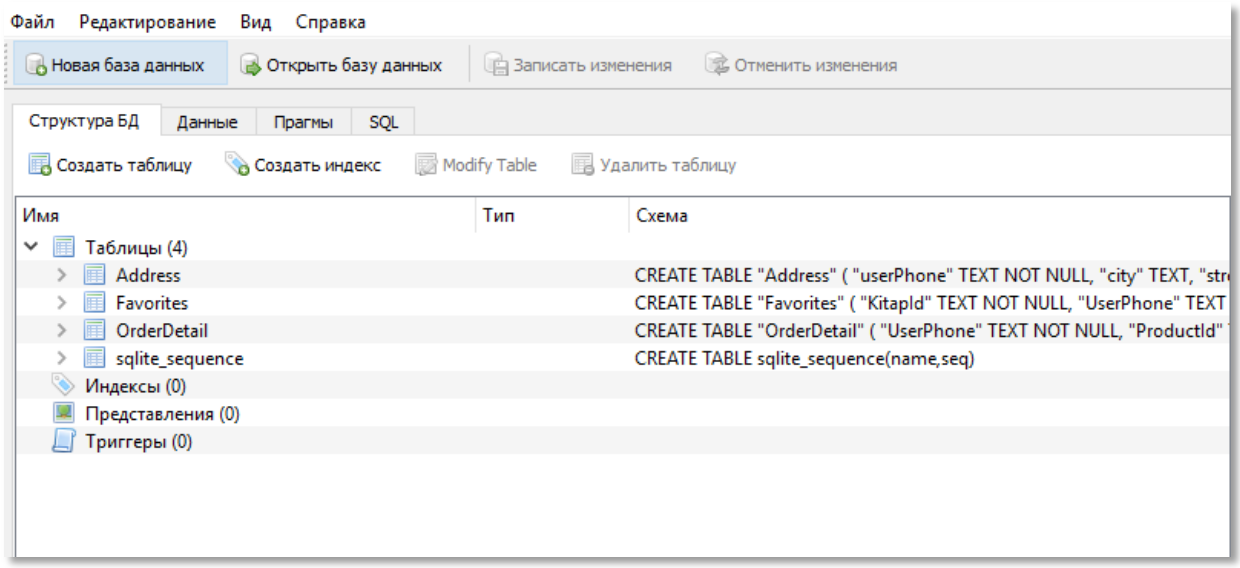
– android.database.sqlite.SQLiteQueryBuilder класы SQL сұрауларын жа- сауға мүмкіндік береді;

– android.database.sqlite.SQLiteOpenHelper класы барлық кестелермен бірге деректер қорын жасауға мүмкіндік береді.

SQLite келесі деректер түрін пайдаланады: INTEGER, REAL, TEXT, BLOB, NUMERIC.

SQLite үшін DB Browser құралын пайдаланып, “KitapDB” деректер қорын және OrderDetail, Favorites, Address кестелерін құрдым. Оны 4.1-суретте көре аламыз.

SQLite үшін DB Browser – бұл SQLite-қа үйлесімді деректер қорының файлдарын жасау, жобалау және өңдеу үшін арналған ашық бастапқы коды бар, жоғары сапалы визуалды құрал. Бұл құралдың интерфейсі қолдануға өте ыңғайлы, ол әзірлеушілерге деректер қорын және ондағы кестелерді тез, әрі оңай құруға мүмкіндік береді.



4.1-сурет – SQLite үшін DB Browser

Кітапты себетке қосу, яғни OrderDetail кестесіне INSERT операциясын орындау коды 4.2-суретте көрсетілген.

Address кестесіндегі мекен-жай тізімін алу 4.3-суретте көрсетілген.

```
// Себетке кітап қосу
public void addToCart(Tapsyrys tapsyrys)
    {
        SQLiteDatabase db = getReadableDatabase();
        String query = String.format("INSERT OR REPLACE INTO
OrderDetail(UserPhone, ProductId, ProductName, Quantity, Price,
Image, Avtor, ISBN) VALUES('%s', '%s', '%s', '%s', '%s', '%s',
'%s', '%s');",
                tapsyrys.getUserPhone(),
                tapsyrys.getTapsyrysId(),
                tapsyrys.getTapsyrysAty(),
                tapsyrys.getSany(),
                tapsyrys.getBagasy(),
                tapsyrys.getImageView(),
                tapsyrys.getAvtor(),
                tapsyrys.getISBN()
        );
        db.execSQL(query);
    }
```

4.2-сурет – Себетке кітап қосу

```
public List<MekenZhai> getMekenZhai(String userPhone)
    {
        SQLiteDatabase db = getReadableDatabase();
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
        String[] sqlSelect = {"ID", "city", "street", "house", "flat",
"userPhone"};
        String sqlTable = "Address";
        qb.setTables(sqlTable);
        Cursor c = qb.query(db, sqlSelect, "userPhone=?", new
String[]{userPhone}, null, null, null);
        final List<MekenZhai> result = new ArrayList<>();
        if (c.moveToFirst())
        {
            do {
                result.add(new MekenZhai(
                        c.getString(c.getColumnIndex("ID")),
                        c.getString(c.getColumnIndex("city")),
                        c.getString(c.getColumnIndex("street")),
                        c.getString(c.getColumnIndex("house")),
                        c.getString(c.getColumnIndex("flat")),
                        c.getString(c.getColumnIndex("userPhone"))
                ));
            }while (c.moveToNext());
        }
        return result;
    }
```

4.3-сурет – Мекен-жай тізімін алу

4.3 Gradle жобасын құрастыру жүйесі

Жобаны құрастыру үшін Gradle пайдаланылды. Gradle – Android Studio әзірлеу ортасына біріктірілген құрастыру құралы. Android Studio әзірлеу ортасында жоба автоматты түрде фондық режимде құрастырылады, бұл құрылым конфигурация файлдарының жиынына негізделеді. Gradle әзірлеушіге қосымшаның тәуелділіктерін оңай басқаруға және оларды репозиторийден орнатуға көмектеседі, сондай-ақ, әр түрлі жинақтарды құруға болады [13].

Build.gradle конфигурация файлы қосымшаның нұсқасы, Android нұсқасының талаптары туралы ақпараттарды қамтиды. 4.4-суретте мобильді қосымшадағы конфигурация файлының бастапқы коды бейнеленген.

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "kz.zhansaya.kitap"
        minSdkVersion 21
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
        }
    }
    productFlavors {
    }
}
}
apply plugin: 'com.google.gms.google-services'
```

4.4-сурет – Build.gradle конфигурациялық файлының коды

Конфигурация файлы сонымен қатар пайдаланылатын бөгде кітапханалардың тізімін қамтиды. 4.5-суретте мобильді қосымшаны әзірлеуде пайдаланылған бөгде кітапханаларды қамтитын конфигурация файлы бейнеленген.

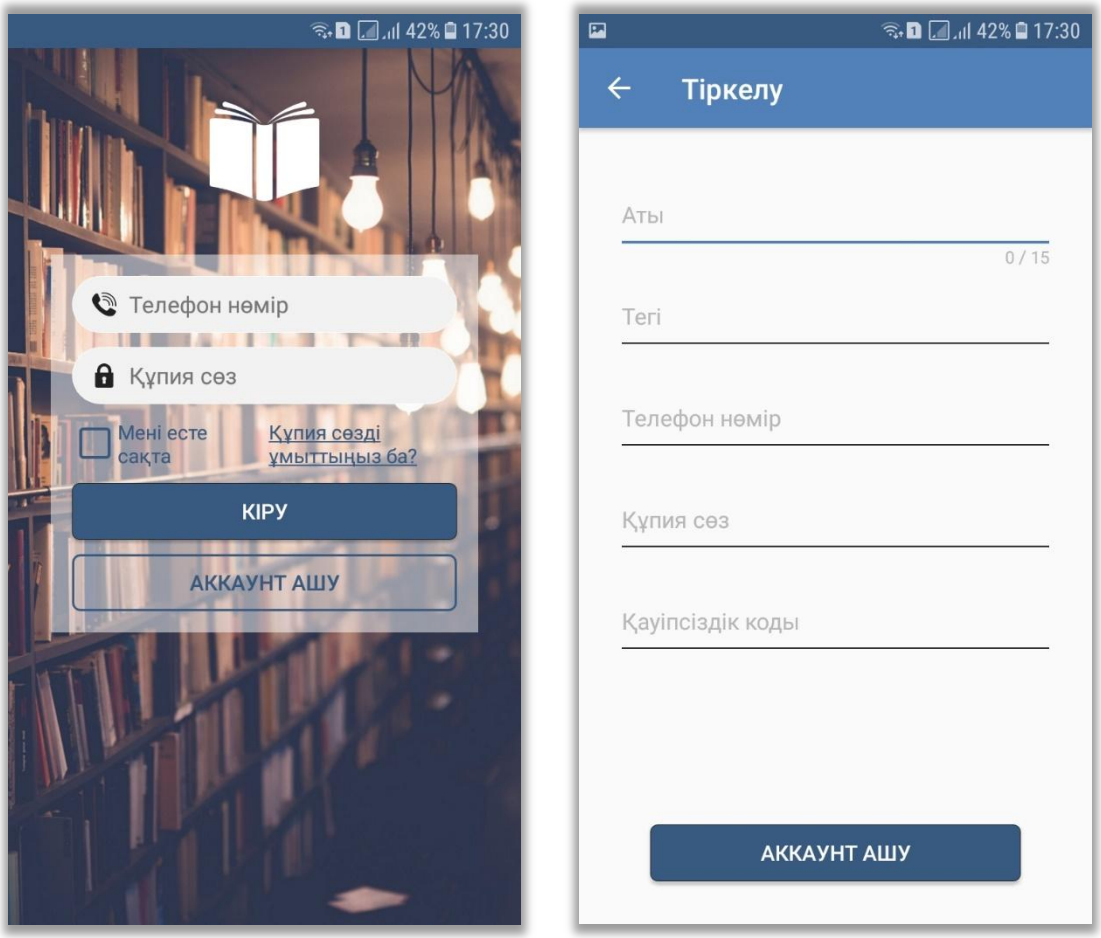
```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support:support-v4:28.0.0'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.rengwuxian.materialedittext:library:2.1.4'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.github.bumptech.glide:glide:4.9.0'
    implementation 'com.firebaseui:firebase-ui-database:1.2.0'
    implementation 'com.cepheuen.elegant-number-button:lib:1.0.2'
    implementation 'com.amulyakhare:com.amulyakhare.textdrawable:1.0.1'
    implementation 'com.github.mancj:MaterialSearchBar:0.7.1'
    implementation 'com.android.support:palette-v7:28.0.0'
    implementation 'io.paperdb:paperdb:2.1'
    implementation 'com.github.rey5137:material:1.2.4'
    implementation 'com.github.andremion:counterfab:1.0.1'
    implementation 'com.google.firebase:firebase-messaging:17.0.0'
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.flaviofaria:kenburnsview:1.0.7'
    implementation 'com.github.florent37:diagonallayout:1.0.7'
    implementation 'com.github.d-max:spots-dialog:1.1@aar'
    implementation 'com.stepstone.apprating:app-rating:2.0.0'
}
```

4.5-сурет – Build.gradle файлына қосылатын кітапханалар тізімі

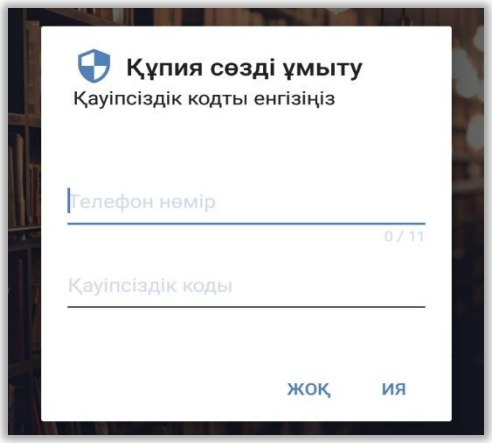
4.4 Қосымша интерфейсі

Әзірленген мобильді қосымша қолданушыға кітап тапсырыстарын жасауға мүмкіндік береді. Қосымшадағы кітаптар категориялар мен жанрларға топтастырылған. Қолданушы деректер қорындағы кез-келген кітап жайында толық ақпаратқа қол жеткізе алады: кітап аты, бағасы, авторы, сипаттамасы, шығарған баспасы, жанры, басқа қолданушылардың кітап жайындағы пікірі және бағасы. Жалпы қосымша интерфейсі өте ыңғайлы және әдемі көрініске ие. Сонымен қатар, қосымша кітап іздеу, кітапқа баға беру және пікір жазу, сүйікті кітаптар қызметтерін ұсынады.

Қосымша іске қосылғанда, пайдаланушы есептік жазбасының деректерін қолдану арқылы авторизациядан өтеді немесе тіркелу шартына сәйкес тіркеу нысанын толтыра отырып, жаңа есептік жазба ашады. 4.6-суретте көрсетілген. Егер барлық енгізілген деректер дұрыс болса, пайдаланушы қосымшаның негізгі бетіне өтеді. Қосымшаға кіргенде, пайдаланушы туралы барлық қажетті ақпарат одан әрі операциялар жасау үшін сақталады.

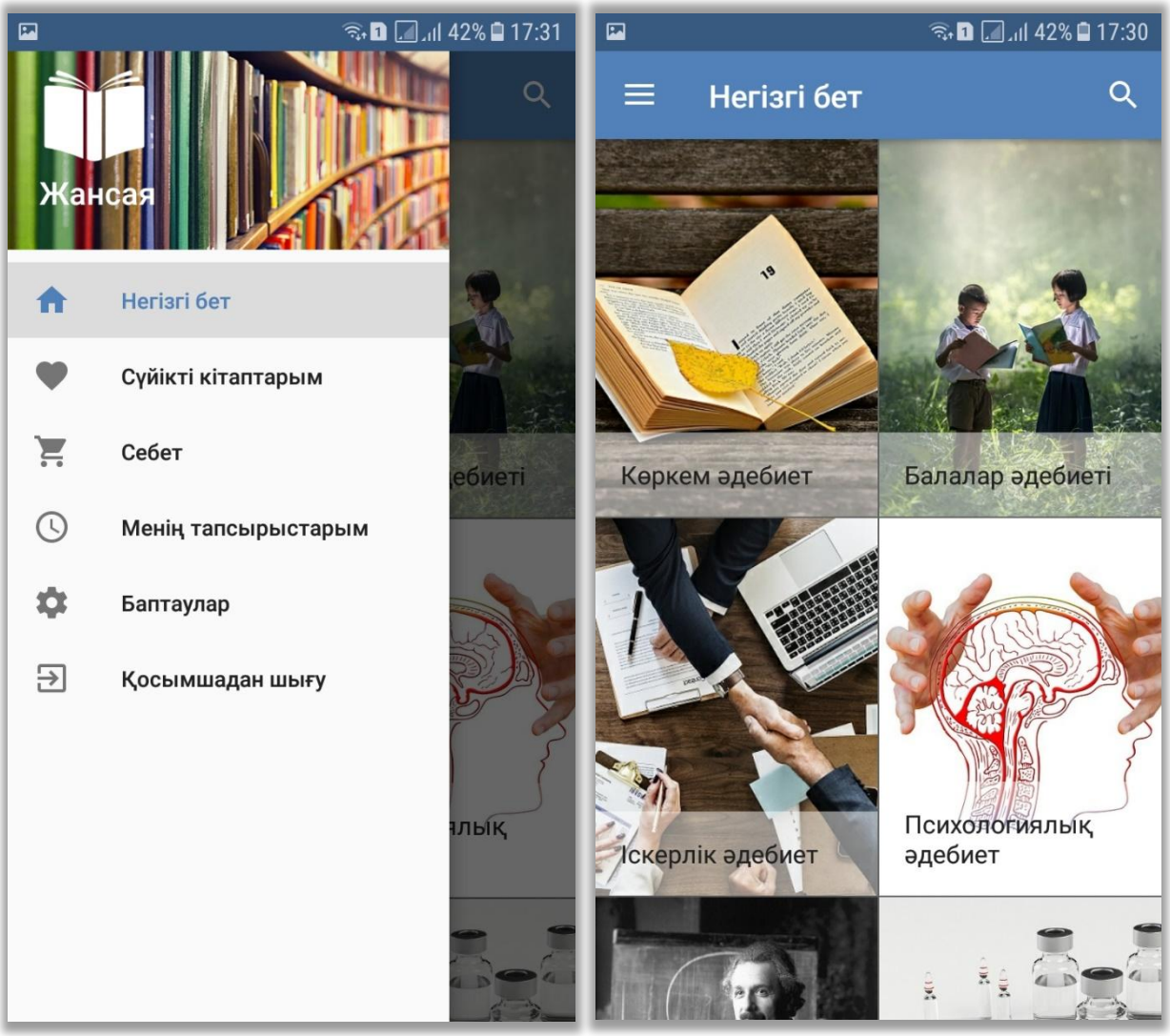


4.6-сурет – Қосымшаға кіру және тіркелу беттері



4.7-сурет – Құпия сөзді ұмыту терезесі

Қосымшаның негізгі бетінде категориялар тізімі көрсетіледі, категория кескіннен және тақырыптан тұрады. Оны 4.8-суретте көре аламыз.



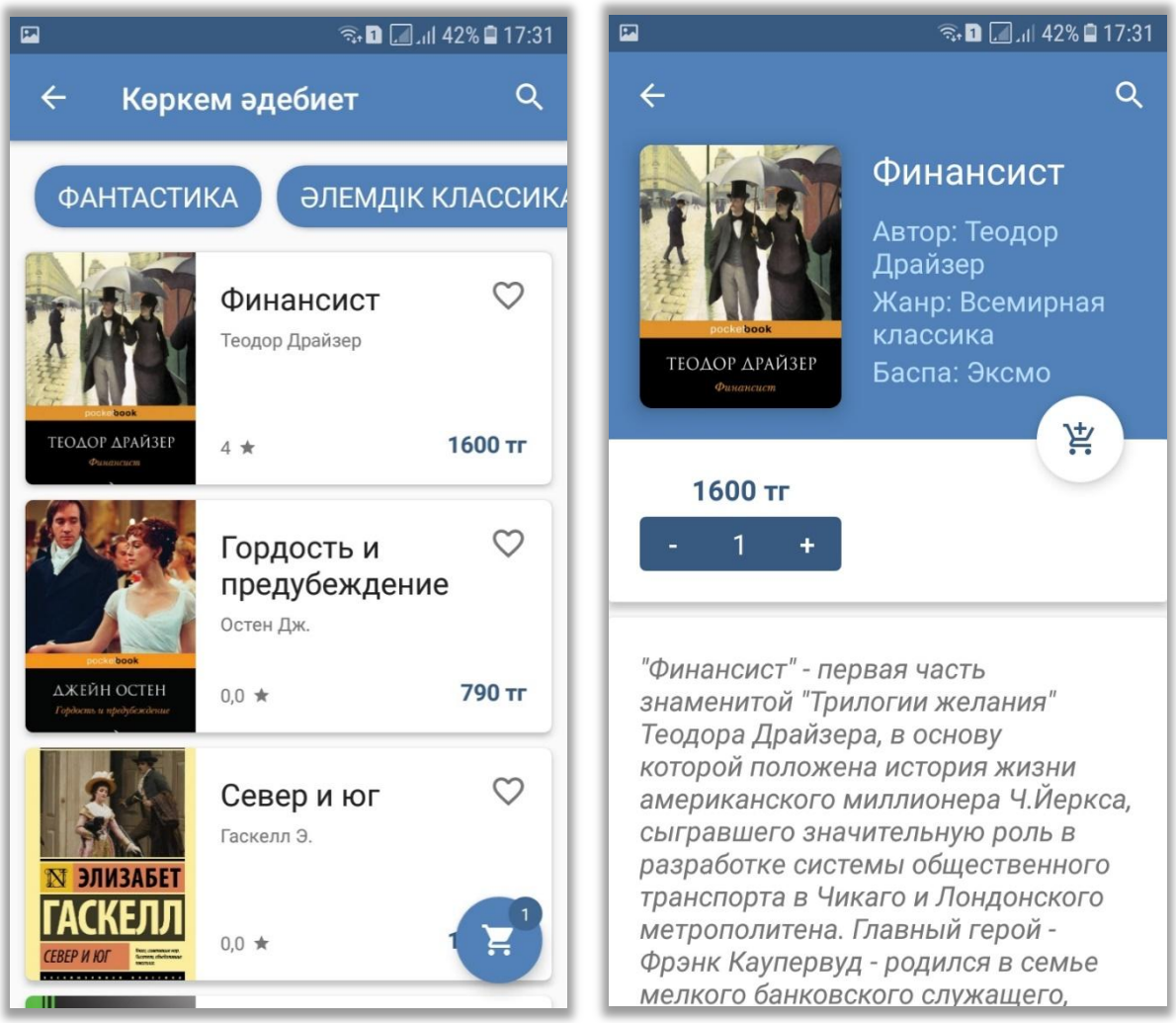
4.8-сурет – Навигация мәзірі және негізгі бет

Қосымшаны навигациялау үшін бүйірлік мәзір қолданылады. Бүйірлік мәзір қосымшаның жоғарғы сол жақ бұрышындағы батырманы басқанда немесе бүйірлік свайп арқылы шақырылады. Әрбір мәзір элементі өз функционалдығын қамтамасыз етеді және ол өзіндік атау мен белгішеден тұрады. Бүйірлік мәзір келесі элементтерді қамтиды: негізгі бет, сүйікті кітаптарым, себет, менің тапсырыстарым, баптаулар, қосымшадан шығу.

Категорияны таңдау арқылы сол категорияға сәйкес кітаптар және жанрлар тізімін алуға болады . 4.9-суретте көрсетілген. Жанрлар көлденең, ал кітаптар тігінен орналасқын RecyclerView виджетінде орналасқан. Мұндағы

тігінен орналасқан RecyclerView, яғни кітаптар тізімі келесі деректерден тұрады: кітаптың кескіні, атауы, авторы, бағасы, жұлдызша саны және лүпіл. Кітап кескінін серверден жүктеу үшін Glide кітапханасы пайдаланылды. Glide кітапханасы басқа танымал Picasso кітапханасының ең жақын бәсекелесі болып табылады және де желіден, ресурстардан немесе файлдық жүйеден кескіндерді асинхронды жүктеуге, оларды кэштеуге және бейнелеуге арналған. Бұл кітапханалардың синтаксисі және жұмыс принципі өте ұқсас.

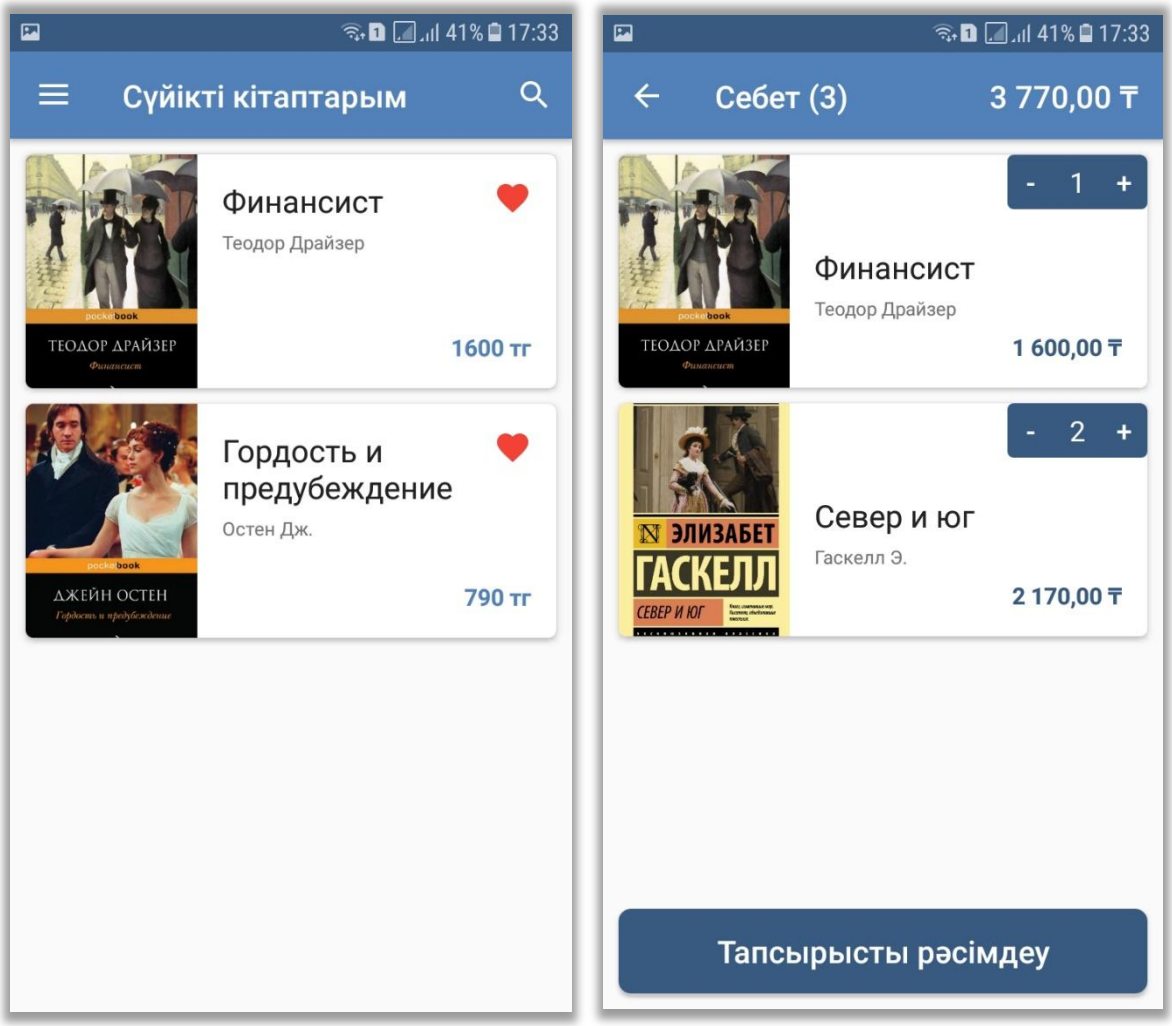
Тігінен орналасқын RecyclerView элементін, яғни кітапты басу арқылы кез келген кітап жайында толық ақпарат аламыз. 4.9-суретте осы туралы бейнеленген, сондай-ақ кітапты себетке қосу үшін “Себетке қосу” батырмасын басамыз.



4.9 – сурет – Кітаптар тізімі және кітап туралы ақпарат беттері

Қосымшада ұнаған кітаптарға лүпіл басып, “Сүйікті кітіптар” тізімін жасауға болады. Ал, кітапқа тапсырыс беру үшін алдын ала қажетті кітаптарды себетке жинау қажет. Сүйікті кітаптар және себеттегі кітаптар SQLite деректер қорында сақталады. Сонымен қатар, мобильді қосымша себеттен кітаптарды алуға және олардың санын өзгертуге мүмкіндік береді. Кітап санын өзгерту

үшін Elegant Number Button виджетін пайдаландым. Бұл – көбейту және азайту батырмалары бар санауышты іске асыруға арналған қарапайым кітапхана [14]. Себеттегі кітаптардың жалпы саны мен бағасы Toolbar виджетінде орналасқын және де бұл өз кезегінде қоданушылар үшін өте ыңғайлы.

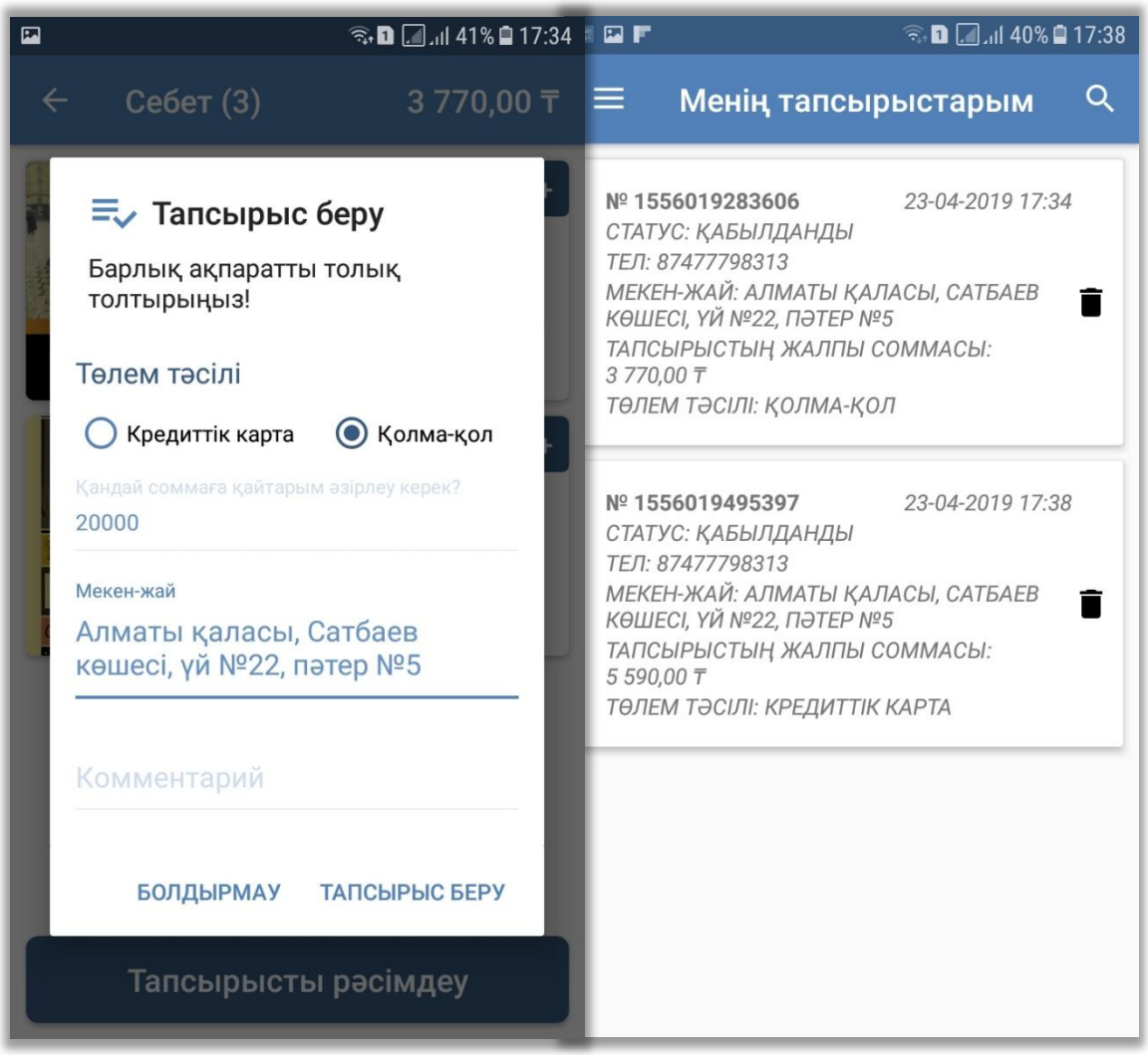


4.10 – сурет – Сүйікті кітаптарым және Себет беттері

“Тапсырысты рәсімдеу” батырмасын басу арқылы тапсырыс жасалады. Қолданушы батырманы басқаннан кейін қосымшада “Тапсырыс беру” диалогтық терезесі ашылады. 4.11-суретте көрсетілген. Мұнда қолданушыға екі түрлі төлем тәсілдері ұсынылады: кредиттік карта және қолма-қол төлем тәсілдері. Егер қолма-қол төлем тәсілі таңдалған жағдайда, тапсырыс беруші міндетті түрде қайтарым үшін нақты ақша сомасын тиісті бөлімге енгізу керек. Тапсырыс сәтті жасалуы үшін барлық ақпараттар толық толтырылуы қажет, әсіресе, мекен-жай міндетті түрде толтырылуы тиіс. Ал, комментарийді жазу міндетті емес. Тапсырыс сәтті түрде жіберілгеннен соң, себет автоматты түрде тазартылады.

Қосымша қолданушыларғы тапсырыс статусын бақылауға мүмкіндік береді. Ол үшін “Менің тапсырыстарым” бетіне өту қажет. Бұл бетте тапсырыс тарихы көрсетіледі: тапсырыстың жасалған күні, орындалу статусы, жалпы

бағасы, тапсырыс берілген кітаптар тізімі, төлем тәсілі. Тапсырыс статусы өзгерген жағдайда, қолданушыға бірден хабарлама келеді. Хабарлама жіберуді

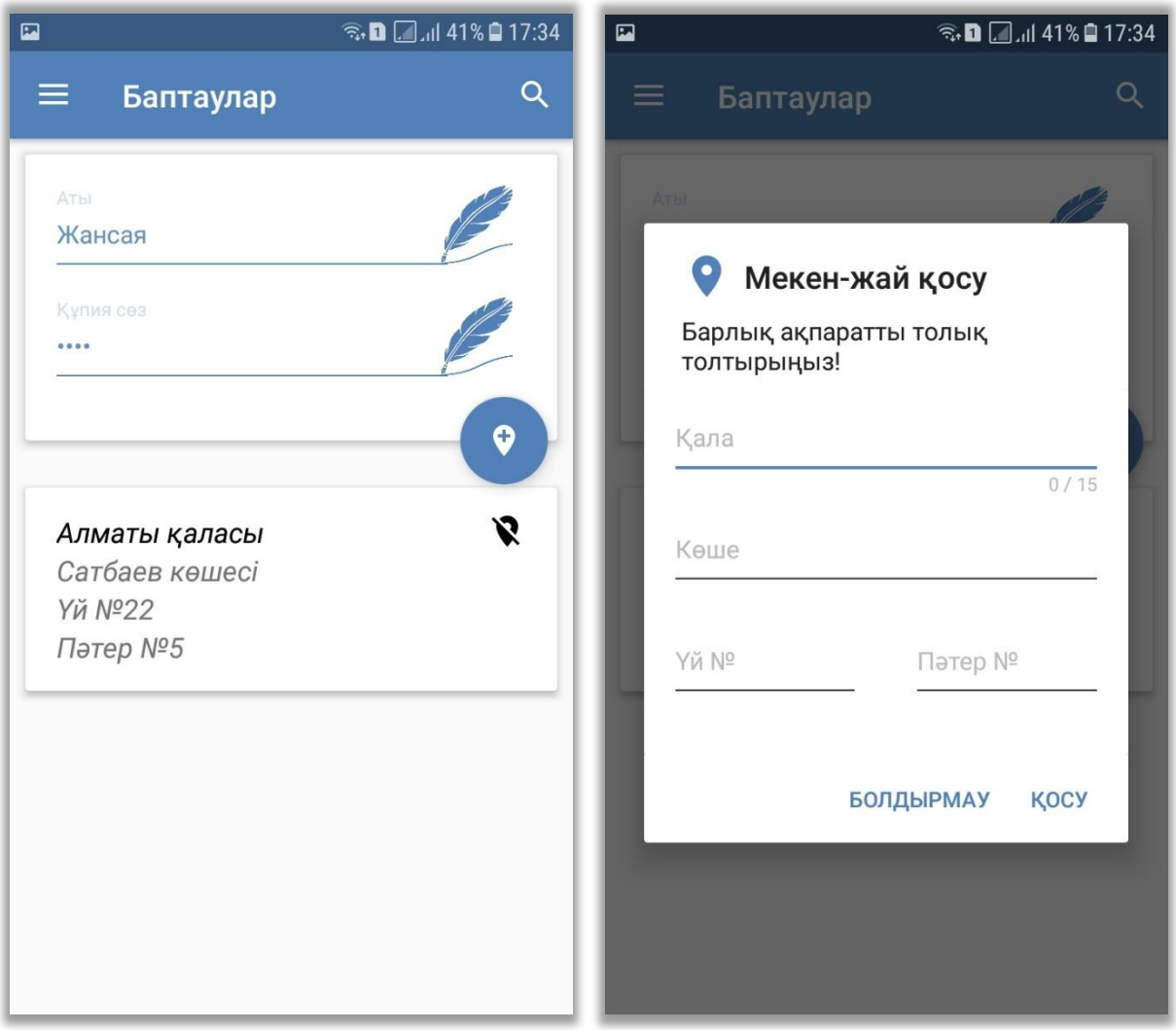


жүзеге асыру үшін FirebaseMessaging қызметін қолдандым.

4.11 – сурет – “Тапсырыс беру” диалогтық терезесі және Менің тапсырыстарым беті

Қолданушылар “Баптаулар” бетіне кіріп, өздерінің жеке деректерін өзгерте алады: есімін және құпия сөзін. 4.12-сурет осы өзгертулер жайлы. Сонымен қатар, қосымша қолданушыларға құрылғы жадысына бірнеше мекен-жайларды сақтауға мүмкіндік береді. Тапсырыс беру кезінде осы алдын-ала сақталған мекен-жайлардың бірі таңдалады.

Қосымшадағы барлық мекен-жайлар SQLite деректер қорындағы Address кестесінде сақталады. Мекен-жай қосу үшін шақырылған диалогтік терезедегі барлық пункттер толық толтырылуы қажет.



4.12-сурет – Баптаулар беті және Мекен-жай қосу терезесі

Қосымшада кітаптарды бағалау қызметін жүзеге асыру үшін AppRating кітапханасы қолдану 4.13-суретте көрсетілген [15]. Бұл кітапхананы қолдану үшін Gradle файлына келесі жолды қосу қажет: compile 'com.stepstone.apprating:app-rating:2.3.1'. Кітапханамен қолдау көрсетілетін функциялар:

– диалог фоны, жұлдызшалары, тақырыбы, сипаттамасы үшін таңдамалы стильдер;

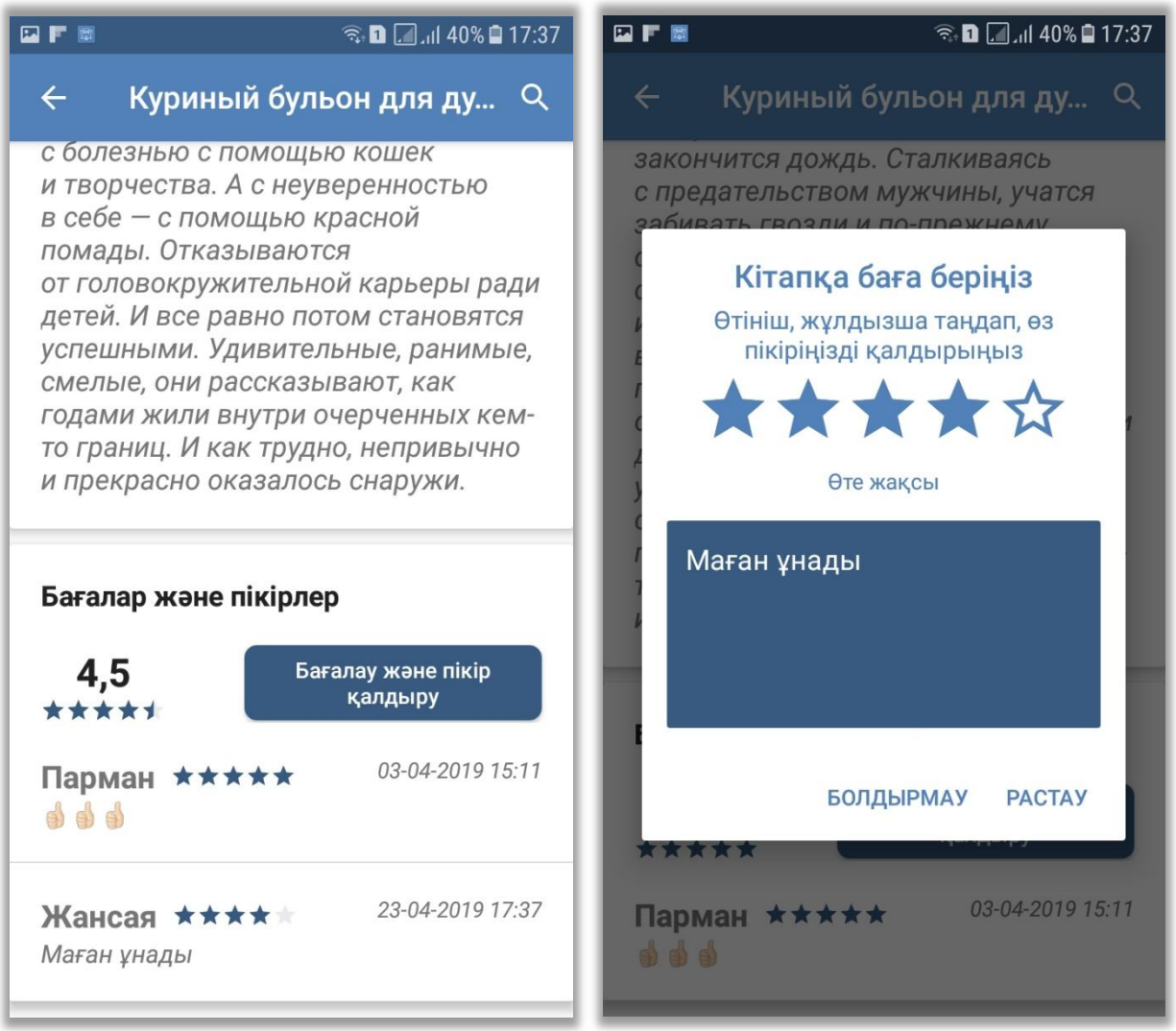
– облыстың пайдаланушы рейтингі (жұлдыздар саны);

– әрбір жазбаны сипаттайтын рейтинг жолының астындағы жазбаның сипаттамасы;

– пайдаланушы диалогының тақырыбын, сипаттамасын және кеңестерін анықтау;

– оң, теріс және бейтарап батырмалары үшін мәтінді анықтау;

– кіру / шығу терезесін анимациялау.



4.13-сурет – Кітапқа баға беру диалогы

ҚОРЫТЫНДЫ

Бұл дипломдық жұмыста кітап сатуға арналған Android қосымшасын әзірлеу бойынша шешім ұсынылды. Android қосымшаларын әзірлеу технологияларына талдау жасалып, нәтижесінде Firebase платформасы қолданылды.

Кітап сату және жеткізу үшін ұқсас шешімдерді іздеу бойынша зерттеу жүргізілді, олардың негізгі сипаттамалары мен мақсаты қарастырылды, сондай-ақ принципті кемшіліктер анықталды.

Бұл дипломдық жұмыста әзірленетін сервистің егжей-тегжейлі сипаттамасы берілді, әзірленетін жүйеге функционалдық және функционалды емес талаптар тұжырымдалды. Жүйемен өзара іс-қимыл жасайтын негізгі актерлер және жүйемен жасауға болатын әрекеттер анықталды. Осы деректер негізінде прецеденттер диаграммасы жасалды,сондай-ақ диаграммада көрсетілген әрбір актердің және прецеденттің егжей-тегжейлі сипаттамасы берілді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Вигерс К.И. Разработка требований к программному обеспечению. – М.: Русская Редакция, 2004. – 9с.

2 Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. – М.: ДМК Пресс, 2006. – 281 с.

3 Barr J. AWS Mobile Hub – Build, Test, and Monitor Mobile Applications [Электронды ресурс] – Қатынау режимі: https://aws.amazon.com/ru/blogs/aws/aws-mobile-hub-build-test-and-monitor-mobile-applications/, тегін.

4 Упрощаем работу с CloudKit, или синхронизация в духе Zen [Электронды ресурс] – Қатынау режимі: https://habr.com/ru/company/everydaytools /blog / 326050/, тегін.

5 Facebook Parse now lets you easily deploy mobile apps to Heroku [Электронды ресурс] – Қатынау режимі: https://venturebeat.com/2015/10/22/facebook -parse -now-lets-you-easily-deploy-apps-to-heroku/, тегін.

6 Материал из Национальной библиотеки им. Н. Э. Баумана [Электронды ресурс] – Қатынау режимі: https://ru.bmstu.wiki/Firebase, тегін.

7 Makan K., Bown S.A. Firebase Essentials – Android Edition. – C.: CreateSpace, 2017. – 9 с.

8 Архитектура мобильного клиент-серверного приложения [Электронды ресурс] – Қатынау режимі: https://habr.com/ru/post/246877/, тегін.

9 Изучаем Retrofit 2 приложения [Электронды ресурс] – Қатынау режимі: https://habr.com/ru/post/314028/, тегін.

10 Дейтел П., Дейтел Х., Дейтел Э., Моргано М. Android для программистов: создаем приложения. – СПб.: Питер, 2013. – С. 349 – 350.

11 Paper is a fast NoSQL [Электронды ресурс] – Қатынау режимі: https://github. com/pilgr/Paper, тегін.

12 Коматинени С., Маклин Д. Android 4 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов. – М.: Вильямс, 2012. – 98 с.

13 Gradle құрастыру құралының ресми веб-сайты [Электронды ресурс] – Қатынау режимі: https://docs.gradle.org/current/userguide/userguide.html#getting-started, тегін.

14 Elegant Number Button кітапханасының ресми веб-саты [Электронды ресурс] – Қатынау режимі: http://www.cepheuen.com/, тегін.

15 AppRating кітапханасының ресми веб-саты [Электронды ресурс] – Қатынау режимі: https://github.com/stepstone-tech/android-material-app-rating, тегін.

А қосымшасы
(міндетті)

Бағдарламаның мәтіні

1.MainActivity.java-Бағдарламаның бастапқы беті

```
package kz.zhansaya.kitap;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.WindowManager;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.rengwuxian.materialedittext.MaterialEditText;
import com.rey.material.widget.CheckBox;
import io.paperdb.Paper;
import kz.zhansaya.kitap.Model.Koldanushy;
import kz.zhansaya.kitap.Zhalpy.Zhalpy;
public class MainActivity extends AppCompatActivity {
AutoCompleteTextView telefonNom, kupiyaSoz;
Button kiru, tirkelu;
CheckBox ckbRemember;
FirebaseDatabase database;
DatabaseReference table_koldanushy;
TextView textView;
 @Override
protected void onCreate(Bundle savedInstanceState) {
```

А қосымшасының жалғасы

```
setContentView(R.layout.activity_main);
telefonNom = (AutoCompleteTextView) findViewById(R.id.telefon);
kupiyaSoz = (AutoCompleteTextView) findViewById(R.id.kupiya_soz);
kiru = (Button) findViewById(R.id.kiru);
tirkelu = (Button) findViewById(R.id.tirkelu);
ckbRemember = (CheckBox) findViewById(R.id.ckbRemember)
textView = (TextView) findViewById(R.id.kauipKod);
 //Init Paper
Paper.init(this);
database = FirebaseDatabase.getInstance();
table_koldanushy = database.getReference("Koldanushy");
textView.setOnClickListener(new View.OnClickListener() {
@Override
Public void onClick(View v) {
showForgotPwdDialog();
}
});
kiru.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
if (Zhalpy.isConnectedToInternet(getBaseContext())){
 //User & password saktau
if (ckbRemember.isChecked())
{
Paper.book().write(Zhalpy.USER_KEY, telefonNom.getText().toString());
Paper.book().write(Zhalpy.PWD_KEY, kupiyaSoz.getText().toString());
}
final ProgressDialog mDialog = new ProgressDialog(MainActivity.this);
mDialog.setMessage("Күте тұрыңыз...");
mDialog.show();
table_koldanushy.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
//Koldanushy bazada bar jogin tekseru
if (dataSnapshot.child(telefonNom.getText().toString()).exists()) {
//Koldanushy turaly akparat alu
mDialog.dismiss();
Koldanushy koldanushy =
dataSnapshot.child(telefonNom.getText().toString()).getValue(Koldanushy.class);
koldanushy.setTelefon(telefonNom.getText().toString());
if (koldanushy.getKupiyaSoz().equals(kupiyaSoz.getText().toString())) {
```

А қосымшасының жалғасы

```
Intent homeIntent = new Intent(MainActivity.this, Home.class);
Zhalpy.agymdagyKoldanushy = koldanushy;
startActivity(homeIntent);
finish();
table_koldanushy.removeEventListener(this);
} else {
Toast.makeText(MainActivity.this, "Құпия сөз қате!",
Toast.LENGTH_SHORT).show();
}} else {
mDialog.dismiss();
Toast.makeText(MainActivity.this, "Бұл қолданушы деректер қорында жоқ!",
Toast.LENGTH_SHORT).show();
}}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
});}
else {
Toast.makeText(MainActivity.this, "Өтініш, интернет байланысын тексеріңіз!",
Toast.LENGTH_SHORT).show();
return;
}});
tirkelu.setOnClickListener(new View.OnClickListener() {
 @Override
public void onClick(View view) {
Intent tirkelu = new Intent(MainActivity.this, Tirkelu.class);
startActivity(tirkelu);
});
String koldanushy = Paper.book().read(Zhalpy.USER_KEY);
String pwd = Paper.book().read(Zhalpy.PWD_KEY);
if (koldanushy != null && pwd != null)
{
if (!koldanushy.isEmpty() && !pwd.isEmpty())
{
login(koldanushy, pwd);
}}}
private void showForgotPwdDialog() {
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Құпия сөзді ұмыту");
builder.setMessage("Қауіпсіздік кодты енгізіңіз");
LayoutInflater inflater = this.getLayoutInflater();
View view = inflater.inflate(R.layout.kupiya_soz, null);
```

А қосымшасының жалғасы

```
builder.setView(view);
builder.setIcon(R.drawable.ic_security_black_24dp);
final MaterialEditText telefon  = (MaterialEditText)
view.findViewById(R.id.telefon);
final MaterialEditText secureCode  = (MaterialEditText)
view.findViewById(R.id.kauipsizKod);
builder.setPositiveButton("Ия", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
table_koldanushy.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
Koldanushy koldanushy = dataSnapshot.child(telefon.getText().toString())
.getValue(Koldanushy.class);
if (koldanushy.getKauipsizKod().equals(secureCode.getText().toString()))
Toast.makeText(MainActivity.this, "Сіздің құпия сөзіңіз:
"+koldanushy.getKupiyaSoz(), Toast.LENGTH_LONG).show();
else
Toast.makeText(MainActivity.this, "Қауіпсіздік коды қате енгізілді!",
Toast.LENGTH_SHORT).show();
}
@Override
public void onCancelled(DatabaseError databaseError) {
}}) ;}});
 builder.setNegativeButton("Жоқ", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
}});
builder.show();
}
private void login(final String telefon, final String pwd) {
if (Zhalpy.isConnectedToInternet(getBaseContext())){
final ProgressDialog mDialog = new ProgressDialog(MainActivity.this);
mDialog.setMessage("Күте тұрыңыз...");
mDialog.show();
table_koldanushy.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
//Koldanushy bazada bar jogin tekseru
 if (dataSnapshot.child(telefon).exists()) {
//Koldanushy turaly akparat alu
```

А қосымшасының жалғасы

```
mDialog.dismiss();
Koldanushy koldanushy = dataSnapshot.child(telefon).getValue(Koldanushy.class);
koldanushy.setTelefon(telefon);
if (koldanushy.getKupiyaSoz().equals(pwd)) {
Intent homeIntent = new Intent(MainActivity.this, Home.class);
Zhalpy.agymdagyKoldanushy = koldanushy;
startActivity(homeIntent);
finish();
} else {
Toast.makeText(MainActivity.this, "Құпия сөз қате!",
Toast.LENGTH_SHORT).show();
}} else {
mDialog.dismiss();
Toast.makeText(MainActivity.this, "Бұл қолданушы деректер қорында жоқ!",
Toast.LENGTH_SHORT).show();
}}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
});}
else {
Toast.makeText(MainActivity.this, "Өтініш, интернет байланысын тексеріңіз!",
Toast.LENGTH_SHORT).show();
return;
}}}
```

2.main_activity.xml файлы

```
<?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:focusable="true"
    android:focusableInTouchMode="true"
    tools:context=".MainActivity">
    <ImageView
        android:scaleType="centerCrop"
        android:src="@drawable/bg"
```

А қосымшасының жалғасы

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        />
<ImageView
        android:layout_marginTop="30dp"
        android:id="@+id/logo"
        android:scaleType="fitCenter"
        android:src="@drawable/ic_book_cover"
        android:layout_width="match_parent"
        android:layout_marginBottom="30dp"
        android:layout_height="85dp" />
    <LinearLayout
        android:layout_below="@+id/logo"
        android:layout_centerInParent="true"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ScrollView
            android:id="@+id/scroll"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <LinearLayout
                android:paddingTop="15dp"
                android:paddingBottom="15dp"
                android:layout_marginStart="30dp"
                android:layout_marginEnd="30dp"
                android:background="@drawable/rectangle"
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <AutoCompleteTextView
                    android:id="@+id/telefon"
                    android:singleLine="true"
                    android:imeOptions="actionNext"
                    android:layout_width="match_parent"
                    android:layout_height="40dp"
                    android:hint="Телефон нөмір"
                    android:text="87477798313"
                    android:inputType="phone"
                    android:textColorHint="#757575"
                    android:textColor="#212121"
```

А қосымшасының жалғасы

```
        android:textSize="16sp"
            android:drawablePadding="5dp"
            android:paddingStart="10dp"
            android:drawableLeft="@drawable/ic_call"
           android:background="@drawable/rounded"
            android:layout_marginStart="15dp"
            android:layout_marginEnd="15dp"
            android:layout_marginBottom="10dp"/>
          <AutoCompleteTextView
            android:text="1234"
            android:singleLine="true"
            android:imeOptions="actionDone"
            android:id="@+id/kupiya_soz"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:hint="Құпия сөз"
            android:textColorHint="#757575"
            android:textColor="#212121"
            android:textSize="16sp"
            android:inputType="textPassword"
            android:drawablePadding="5dp"
            android:paddingStart="10dp"
            android:drawableLeft="@drawable/ic_lock"
            android:background="@drawable/rounded"
            android:layout_marginStart="15dp"
            android:layout_marginEnd="15dp"
            android:layout_marginBottom="10dp"/>
          <LinearLayout
            android:orientation="horizontal"

android:weightSum="2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <com.rey.material.widget.CheckBox
              android:layout_marginStart="15dp"
              android:layout_marginEnd="15dp"
              android:layout_marginBottom="10dp"
              android:id="@+id/ckbRemember"
              android:textColor="@color/colorPrimaryDark"
              app:cbd_strokeColor="@color/colorPrimaryDark"
              app:cbd_tickColor="@android:color/white"
```

А қосымшасының жалғасы

```
android:textStyle="normal"
                android:gravity="center_vertical"
                android:text="Мені есте сақта"
                style="@style/Material.Drawable.CheckBox"
                android:layout_weight="0.90"
                android:layout_width="0dp"
                android:layout_height="wrap_content" />
            <TextView
                android:layout_marginStart="15dp"
                android:layout_marginEnd="15dp"
                android:id="@+id/kauipKod"
                android:textColor="@color/colorPrimaryDark"
                android:text="@string/kupiyaSoz"
                android:layout_weight="1.10"
                android:layout_width="0dp"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <Button
            android:id="@+id/kiru"
            android:layout_marginBottom="10dp"
            android:clickable="true"
            android:text="Кіру"
            android:textSize="15dp"
            android:layout_gravity="center"
            android:textStyle="normal"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:layout_marginStart="15dp"
            android:layout_marginEnd="15dp"
            android:background="@drawable/kiru_batyrmasy"
            android:textColor="#FFFFFF"/>
        <Button
            android:id="@+id/tirkelu"
            android:clickable="true"
            android:text="Аккаунт ашу"
            android:textSize="15dp"
            android:layout_gravity="center"
            android:textStyle="normal"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:layout_marginStart="15dp"
```

```
  android:layout_marginEnd="15dp"
                android:background="@drawable/akkaunt_ashu"
                android:textColor="@color/colorPrimaryDark"/>
          </LinearLayout>
       </ScrollView>
 </LinearLayout>
    <!-- <Button
        android:layout_marginBottom="15dp"
        android:clickable="true"
        android:textSize="15dp"
        android:textStyle="normal"
        android:textColor="#f2f2f2"
        android:text="Аккаунт ашу"
        android:layout_marginTop="10dp"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:background="@drawable/akkaunt_ashu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />-->
</RelativeLayout>
```

3.Home.java Activity беті

```
package kz.zhansaya.kitap;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.view.View;
import android.support.design.widget.NavigationView;
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 android.widget.TextView;
```

А қосымшасының жалғасы

```
import com.google.firebase.iid.FirebaseInstanceId;
import io.paperdb.Paper;
import kz.zhansaya.kitap.Fragments.Baptaular;
import kz.zhansaya.kitap.Fragments.FavFragment;
import kz.zhansaya.kitap.Fragments.KatFragment;
import kz.zhansaya.kitap.Fragments.TapsyrysFragment;
import kz.zhansaya.kitap.Model.Token;
import kz.zhansaya.kitap.Zhalpy.Zhalpy;
public class Home extends AppCompatActivity
implements NavigationView.OnNavigationItemSelectedListener {
public static TextView koldanushyAty;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_home);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
Paper.init(this);
//      FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//      fab.setOnClickListener(new View.OnClickListener() {
//          @Override
//          public void onClick(View view) {
//              Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
//                      .setAction("Action", null).show();
//          }
//      });
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();
NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
View headerView = navigationView.getHeaderView(0);
koldanushyAty = (TextView) headerView.findViewById(R.id.koldanushy_aty);
koldanushyAty.setText(Zhalpy.agymdagyKoldanushy.getAty());
 //Select Home by default
navigationView.setCheckedItem(R.id.nav_camera);
```

А қосымшасының жалғасы

```
displaySelectedFragment(fragment);
updateToken(FirebaseInstanceId.getInstance().getToken());
}
private void updateToken(String token) {
FirebaseDatabase db = FirebaseDatabase.getInstance();
DatabaseReference tokens = db.getReference("Tokens");
Token data = new Token(token, false);
tokens.child(Zhalpy.agymdagyKoldanushy.getTelefon()).setValue(data);
}
@Override
public void onBackPressed() {
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
if (drawer.isDrawerOpen(GravityCompat.START)) {
drawer.closeDrawer(GravityCompat.START);
} else {
super.onBackPressed();
}}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
 // Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.home, 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.search) {
startActivity(new Intent(Home.this, SearchActivity.class));
}
return super.onOptionsItemSelected(item);
}
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
// Handle navigation view item clicks here.
int id = item.getItemId();
Fragment fragment = null;
```

А қосымшасының жалғасы

```
if (id == R.id.nav_camera) {
fragment = new KatFragment();
displaySelectedFragment(fragment);
}else if (id == R.id.nav_slideshow) {
fragment = new FavFragment();
displaySelectedFragment(fragment);
} else if (id == R.id.nav_manage) {
Intent sebetIntent = new Intent(Home.this, Sebet.class);
startActivity(sebetIntent);
}
else if (id == R.id.tapsyrys) {
fragment = new TapsyrysFragment();
displaySelectedFragment(fragment);
}
else if (id == R.id.baptaular) {
fragment = new Baptaular();
displaySelectedFragment(fragment);
}
else if (id == R.id.shygu) {
//Saktalgan koldanushyny zhyoyu
Paper.book().destroy();
Intent kiruIntent = new Intent(Home.this, MainActivity.class);
kiruIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
startActivity(kiruIntent);
}
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}
private void displaySelectedFragment(Fragment fragment) {
FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
fragmentTransaction.replace(R.id.frame, fragment);
fragmentTransaction.commit();
}
}
```

Форматы	Зон	Поз-сы			Саны	Қорытынды

					Дипломдық жоба		Бет
							54
Өлш	Бет	Құжат №	қолы	Күні			

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
СӘТБАЕВ УНИВЕРСИТЕТІ

Қ.И. Сәтбаев атындағы Қазақ Ұлттық техникалық зерттеу университеті

Мамандығы 5В060200 – Ақпараттану

Студент Парман Жансая Құлжанқызы

Тақырыбы: «Android платформасында кітаптарды сату интернет-дүкені»

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Дипломдық жобада қазіргі заманға сай кітап сату дүкеніне арналған Android мобильді қосымшаны құру қарастырылған. Мобильді қосымшаның артықшылығы тұтынушыларға әр түрлі жанрдағы кітаптарды ұсына отырып, тапсырыс жасауға тиімді қызмет көрсетуге мүмкіндік беруінде.

Дипломдық жұмыстың бірінші бөлімі кітап жеткізу қызметімен айналысатын жүйелерді шолуға арналған. Екінші бөлімі мобильді қосымшаға қойылатын функционалдық және функционалды емес талаптарды талдауды қамтиды. Үшінші бөлімде Android операциялық жүйесінде мобильдік қосымшаларды әзірлеу технологиялары қарастырылған. Төртінші бөлімде мобильді қосымшаны іске асыру ұсынылған.

Дипломдық жобаны талдай келе, жұмыстың материалы жүйелі түрде, қойылған талаптарға сай, тақырып төңірегінде жазылғанын айта кетуге керек. Студенттің жұмысын жақсы жасап шыққанын, талдау жағынан басымдылығын байқауға болады. Дипломдық жұмысы арқылы қолданысы кең сала бойынша маңызды жағдайларға мән бере отырып, зерттеу жұмысында жақсы нәтижелер көрсетті.

Қортындылай келгенде, осы зерттеу жұмысын орындау барысында Парман Жансая Құлжанқызы өте жақсы нәтижесімен, еңбектенгені көрінеді.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Парман Жансая Құлжанқызының «Android платформасында кітаптарды сату интернет-дүкені» атты тақырыптағы дипломдық жұмысы қорғауға жіберілді және 5В060200– «Ақпараттану» мамандығы бойынша «Жаратылыстану бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ _____ **Р.С. Алғожаева**
«Программалық инженерия»
кафедрасының сеньор-лекторы «12» 05 2019жыл

Ф ҚазҰТЗУ 704-23. Сын-пікір