

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

Программалық инженерия кафедрасы

Даулетканова Молдир Муратканқызы

Тақырыбы: Дәріханалар бойынша мобильдік қолданба-агрегаторды әзірлеу

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

5B060200 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Алматы 2021


ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

Программалық инженерия кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

ПИ кафедрасының меңгерушісі,  
PhD докторы

 М. Тұрдалыұлы  
«06» 06 2021ж.

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**


Тақырыбы: Дәріханалар бойынша мобильдік қолданба-агрегаторды әзірлеу

5B060200 – «Есептеу техникасы және бағдарламалық камтамасыз ету»  
мамандығы

Орындаған:

Даулетканова Молдир Муратканқызы

Ғылыми жетекші, тех.ғыл.  
магистрі, лектор Баймбетов Д.А.

  
« 04 » 06 2021ж.

Алматы 2021



ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
СӘТБАЕВ УНИВЕРСИТЕТІ

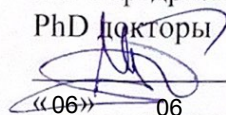
Кибернетика және ақпараттық технологиялар институты

Программалық инженерия кафедрасы

**БЕКІТЕМІН**

ПИ кафедрасының меңгерушісі,

PhD докторы

 М. Тұрдалыұлы

«06» 06 2021 ж.

Дипломдық жоба орындауға

**ТАПСЫРМА**

Білім алушыға Даулетканова Молдир Муратканқызы

Тақырыбы: «Дәріханалар бойынша мобильдік қолданба-агрегаторды әзірлеу»

Университет ректоры бұйрығының № 2131-б « 24 » 11 2021ж. Шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі « 7 » 06 2021 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық құжаттаманы әзірлеу кезінде құрылған жобаның UML диаграммалары, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) мобильді қолданба-агрегаторына жалпы сипаттама беру және ұқсас қосымшаларды талдау;

б) жобаны модельдеу және UML диаграмма құру;

в) пайдаланушы интерфейсін жобалау және дамыту;

г) бағдарламаны құру, кітапханалармен жұмыс жасау, компоненттерді қосу және тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен): презентацияның 24 слайдпен берілген құжат түрінде ұсынылған.



Ұсынылған негізгі әдебиеттер: 10 пайдаланылған әдебиеттер тізімінен.



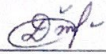
Дипломдық жобаны орындау  
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Диплом жұмысының жоспар құрылымын құру	20.01.2021	жоқ
2. Тапсырма қойылымы және бағдарламау ортасын таңдау	05.02.2021	жоқ
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім беру бойынша есеп беру жазбасын дайындау	10.02.2021	жоқ
4. Дипломның екінші бөлімі – жобалау сызбаларын дайындау.	15.03.2021	жоқ
5. Мобильді қосымшаның UI әзірлеу. Қосымшаны тестілеуден өткізу.	09.04.2021	жоқ
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	06.05.2021	жоқ

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойылған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Марғұлан К., техн. ғыл. магистрі, лектор	06.06.2021	
Бағдарламалық бөлім	Рамазан А.Б., техн. ғыл. магистрі, ассистент	31.05.2021	

Ғылыми жетекші  Д.А. Баймбетов

Тапсырманы орындауға қабылдап алған студент  М.М. Даулетканова

Күні « 06 » 06 2021ж.

## АҢДАТПА

Бұл дипломдық жоба дәріханалардың қызметін мобильдік қолданба-агрегаторын қолдану арқылы жеңілдету мақсатында құрылған.

Онлайн дәріханалардың мобильдік қолданба агрегаторы тұтынушыға қажетті дәрі-дәрмектің тез әрі, дәл қол жетімділігін тексеру, сонымен қатар, шолуларды, жеңілдіктер мен артықшылықтарды салыстыру арқылы қолайлы ұсынысты таңдау мүмкіндігін береді. Сондай-ақ, қосымшада дәріхананың суреті, мекен-жайы, телефон нөмірі, жұмыс режимі туралы толық ақпарат қамтылғандықтан, дәріханалармен байланыс орнату функциясында қарастырылған. Жобалау кезінде пайдаланушы интерфейсі мүмкіндігінше қарапайым және интуитивті түсінікті етіп жасалды.

## АННОТАЦИЯ

Данный дипломный проект создан с целью упрощения деятельности аптек с помощью мобильного приложения-агрегатора.

Агрегатор мобильного приложения онлайн-аптек предоставляет потребителю возможность быстро и точно проверить доступность необходимого лекарства, а также выбрать выгодное предложение, сравнив отзывы, скидки и преимущества. А также, в приложении содержится подробная информация об аптеках, фото, адрес, номер телефона, режим работы, предусмотрена функция связи с аптеками. При проектировании пользовательский интерфейс создавался максимально легким и простым.

## **ANNOTATION**

This graduation project was created to simplify the activities of pharmacies with the help of a mobile aggregator application.

The aggregator of the mobile application of online pharmacies provides the consumer with the opportunity to quickly and accurately check the availability of the necessary medicine, as well as choose a favorable offer by comparing reviews, discounts and benefits. Since the application contains detailed information about the photo of the pharmacy, address, phone number, working hours, the function of communication with pharmacies is provided. When designing, the user interface is designed to be as simple and easy as possible.

## МАЗМҰНЫ

Кіріспе	8
1 Зерттеу бөлімі	9
1.1 Пәндік аймақтың сипаттамасы	9
1.2 Қолданбаны әзірлеу мақсаты	10
1.3 Анықтамалар, терминдер және қысқартулар	10
1.4 Ұқсас қосымшаларды талдау	11
1.4.1 I-teka, Dari.kz бағдарламаларына шолу	11
2 Технологиялық бөлім	13
2.1 Қолданылған құралдарға шолу	13
2.1.1 Flutter SDK	13
2.1.2 Dart бағдарламалау тілі	15
2.1.3 IDE Android Studio	16
2.1.4 AVD Manager	16
3 Жобалау бөлімі	18
3.1 «Дәріханалар бойынша мобильдік қолданба-агрегатор» қосымшаны UML модельдеу тілі көмегімен жобалау	18
3.1.1 UML біріңғай моделдеу тілі	18
3.1.2 Прецеденттер диаграммасы (Use Case diagram)	19
3.1.3 Класстар диаграммасы (Class diagram)	20
3.1.4 Компоненттер диаграммасы (Component diagram)	21
3.1.5 Белсенділік диаграммасы (Activity diagram)	22
3.2 Мобильдік қолданба-агрегаторының қолданушылар интерфейсін әзірлеу	22
Қорытынды	30
Пайдаланылған әдебиеттер тізімі	31
А Қосымшасы. Техникалық тапсырма	32
Б Қосымшасы. Бағдарлама мәтіні	34
Спецификация беті	



## КІРІСПЕ

Қазіргі таңда адамға маңызды заттардың бірі уақыт болып табылады, осыған байланысты мобильді технологиялар адамдардың күш-қуатын, уақытын, энергиясын үнемдеуге барлық мүмкіндікті береді. Үлкен кезектерде тұрып, күту немес көлік кептелістерінде тұру біршама үлкен қиындықтар туғызғандықтан, ғаламтор желісі, ақпараттық жүйелер, мобильді қосымшалар арқылы жұмыстарды бітіру жеңілдей түсті.

Бүгінгі мәдени әлемде ақпараттандыру кезеңі, оның әлеуметтік-тұрмыстық жағдайларына қарамастан, өзінің барлық мүшелерімен бірге өз білімдерін қолдана отырып, олардың алдында үздіксіз пайда болып отыратын мәселелерді шешеді. Мысалы, әртүрлі мобильді қосымшалардың көмегімен сіз үйден шықпастан, дәрі-дәрмектерді сатып ала аласыз, дәрі-дәрмектердің бағаларын салыстыра аласыз немесе фармацевттен кеңес алу мүмкіндігі бар. Бүгінгі күні дәріханалардың мобильді қосымшалары жоғары сұраныста, және өркениетті мемлекеттер арасында үлкен орынға ие. Сондай-ақ, денсаулық сақтау саласындағы негізгі құрылым және экономикалық тұрғыдан тиімді салаларының біріне фармацевтика жатады.

Дипломдық жобаның негізгі мақсаты – дәріхананың мобильді қосымшасын құру және пайдаланушылардың уақытын үнемдеу және қажеттіліктеріне сай қызмет көрсету. Осы мобильді қосымша көмегімен пайдаланушылар қажетті дәрі-дәрмектерді іздеп табумен қатар, жеңілдіктер мен артықшылықтарды салыстыру арқылы қолайлы ұсынысты таңдап алуға, пікір блогын қарауға, дәріханалар туралы толық ақпаратқа қол жеткізу мүмкіндіктері болады.

Сонымен қатар, қазіргі кезде бүкіл әлемде, оның ішінде Қазақстанда коронавирустық эпидемияның өршуі әсерінен, дәріханалардағы дәрі дәрмектің қол жетімділігі халық арасында үлкен алаңдатушылық тудыруда. Бұл дипломдық жоба дәріханалардың қызметін және қолданушылардың мәселелерін мобильдік қолданба-агрегаторын қолдану арқылы жеңілдету үшін таптырмас шешім болып табылады.

Өзіміз көріп тұрғандай көптеген мәселерелердің шешімін мобильді қосымшалардың көмегімен шешуге болады.

## 1 Зерттеу бөлімі

### 1.1 Пәндік аймақтың сипаттамасы

Қазіргі кезде дәріханалардың мобильді қосымшалары қоғамның дамып келе жатқан және өркениетті мемлекеттерінің ішінде өз орнын алып, үлкен жетістіктерге жету үстінде. Сонымен дәріхананың агрегаторлары деген не? Сондай-ақ ол не үшін керек? деген сұрақтарға жауап қарастырайық және пәндік аймаққа қысқаша сипаттама беріп өтейік.

Біздің уақытымызда, нарықта дәріханалардың екі түрі ұсынылған: қарапайым дәріханалар және дәрі-дәрмекті сатуға арналған мобильді қосымшалар. Дәріхананың агрегаторы дегеніміз – бұл қаладағы дәріханалардың мәліметтерін көрсететін сайттар, мобильді қосымшалар. Әрбір агрегатордың жеке мәліметтер базасы бар. Барлық қызметтер шамамен бірдей жұмыс істейді. Сайттарда, мобильді қосымшаларда препараттың атауын енгізуге арналған енгізу орны қарастырылған. Егер іздеу нәтижесі қанағаттанарлық болса, қолданушы тапсырыс берілетін өнімді алуды қалайтын жері бойынша ыңғайлы дәріхананы таңдайды. Броньдаудың ұзақтығы таңдалған дәріханаға байланысты болады.

Көптеген сайттар, мобильді қолданба – агрегаторлар пайдаланушыларға қосымша мүмкіндіктерді ұсынады. Мысалы, препарат туралы толық ақпарат алуға мүмкіндік беретін дәрі-дәрмектердің анықтамалығын алуға болады (құрамы, жанама әсерлері, қолдану әдісі). Сондай-ақ, қосымша агрегаторында дәріхананың мекен-жайы, жұмыс режимі, картадан орналасқан жері мен телефон нөмірі туралы толық ақпараттарды тауып беруді қамтамасыз етеді. Қазір онлайн дәріханалар әлемдік деңгейде даму үстінде болғандықтан, дәріхананың мобильді қосымшасы да күннен-күнге өзгеріп, жақсаруда және оның түрлері де өте көп.

Бұл мобильді қолданба – агрегаторының қолданушыға қай жағынан жәрдем болады, әрі неліктен қолайлы мобильді қосымша болып табылатындығына келетін болсақ:

- жоғарыда айтылып кеткендей ең алдымен мобильді қосымша адамның құндылықтарының бірі уақытты үнемдеуге мүмкіндік береді;
- дәріхананың мобильді қолданба-агрегаторы тәулігіне 24 сағат жұмыс істейді (24/7);
- сондай-ақ, мобильді қосымшада дәріханалардағы бағаларды салыстыра отырып, бағасы жағынан қолайлы ұсынысты таңдай аласыз;
- дәрі-дәрмекті қолданған адамдардың дәрі туралы қалдырып кеткен пікірлерін оқып, дәрі-дәрмектің, басқа да тауарлардың көмегі, әсері бар немесе жоқтығына көз жеткізе аласыз;
- үйде отырып-ақ, сирек кездесетін дәрілерді аздаған уақытта табуға мүмкіндік береді. Яғни бұл мобильді қосымша пайдаланушыға жайлылықты сезінуді қамтамасыз етеді.

## 1.2 Қолданбаны әзірлеу мақсаты

Жұмысты әзірлеу мақсаты – дәріханалардың жұмысын жеңілдететін және қосымша қолданушыларына қолайлы мобильді қосымша түрінде өнім жасау.

Жұмыстың өзектілігі:

- қолданушы үйде отырып-ақ, қажетті ақпаратқа қол жеткізетін;
- дәріхана иелері деректерді уақытылы жаңартып отыратын мүмкіндігі бар мобильді қолданба агрегаторын жобалап шығару болып табылады.

Мұндай бағдарламалық өнімнің артықшылығы – дәріхана иесі мен басқа да қосымша қосымшаларының адами ресурстарын, яғни уақытын, күш-қуатын, энергиясын үнемдеуге мүмкіндік береді.

Қорытынды өнім қолданушыларға келесі мүмкіндіктерді береді:

- авторизациялау және тіркеу мүмкіндігі;
- өзіңіз туралы алғашқы мәліметтерді көрсету;
- дәріханалардағы дәрі-дәрмектің қол жетімділігін тез тексеру;
- шолуларды, жеңілдіктер мен артықшылықтарды салыстыру арқылы қолайлы ұсынысты таңдау;
- дәріхананың суреті, мекен-жайы, телефон нөмірі, жұмыс режимі туралы толық ақпарат алу;
- жеңілдіктер туралы push-хабарламалар алу.

## 1.3 Анықтамалар, терминдер және қысқартулар

Әзірлеу кезінде қолданылатын, сондай-ақ іске асыру тәсіліне және пайдаланылған технологияларына байланысты терминдер 1-кестеде көрсетілген.

### 1-кесте – Терминдер, қысқартулар, және анықтамалар

SDK	Software Development Kit
UI	User Interface (Қолданушы интерфейсі)
AOT	Ahead of Time Compilation (қосымшаны іске қосар алдында компиляциялау)
JIT	Just in Time (қосымшаны іске қосу кезінде компиляциялау)
IDE	Integrated Development Environment
APK	Android Package Kit
ADT	Android Development Tools
UML	Unified Modeling Language

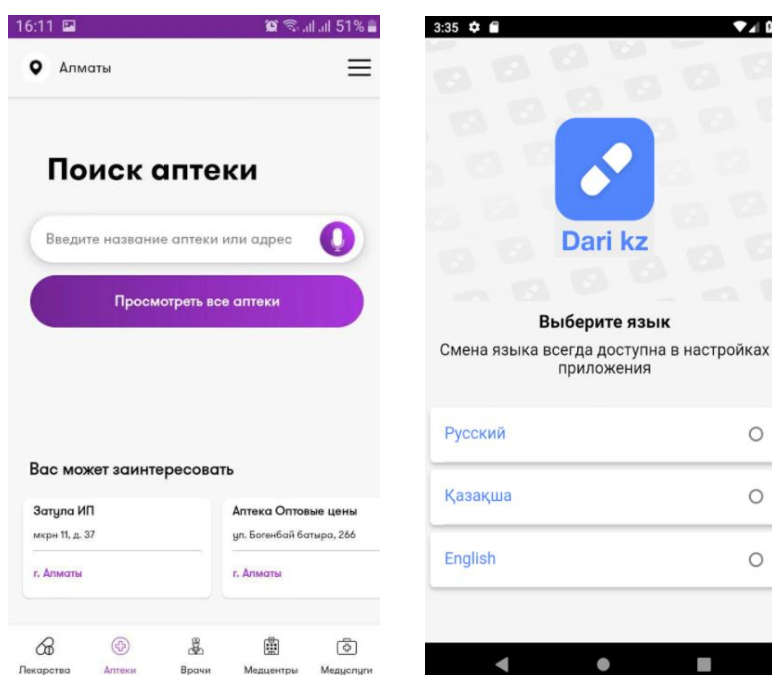
## 1.4 Ұқсас қосымшаларды талдау

### 1.4.1 I-teka, Dari kz бағдарламаларына шолу

Қазақстандағы дәріханалардың қолда бар мобильді қосымшалары бойынша ұқсас қосымшаларға таңдалмалы талдау жүргізілді.

Талдау барысында екі мобильді қосымша таңдалды (1.1-сурет):

- 1-ші мобильдік қосымша – i-teka;
- 2-ші мобильдік қосымша – Dari kz.



#### 1.1-сурет – I-teka және Dari kz мобильді қосымшаларның негізгі беті

«I-teka» мобильді қосымшасы Қазақстанның ақпараттық нарығында 2014 жылдан бастап жұмыс істейді. Мобильді қосымшадан шықпай-ақ, Қазақстан дәріханаларында дәрілердің бар-жоғы туралы ақпаратты табуға және дәріханаға немесе медициналық мекемеге хабарласу мүмкіндігі қарастырылған. Сондай-ақ, пайдаланушының қазіргі орналасқан жерін қолдана отырып, интерактивті навигация сізге жақын дәріхананы немесе медициналық мекемені табуға көмектеседі.

Артықшылықтары:

- қосымша тегін орнатылады және пайдаланылады;
- ынғайлы және минималистік интерфейс;
- қосымша тек Қазақстанда жұмыс істейді;

Кемшіліктері:

- ынғайсыз навигация;
- көп дәрі-дәрмектің нұсқаулықтары жоқ;
- деректер нақтылыққа сәйкес келмейді;



- дәрі-дәрмектің бағасы жайлы ақпарат ұзақ уақыт жаңартылмайды;
- дәрі-дәрмектерді немесе картадан дәріханаларды іздеген кезде қосымша автоматты түрде жабылады. [8]

"Dari kz" мобильді қосымшасы ҚР тұрғындарына Android және iOS платформасында тегін негізде қол жетімді. Осы мобильдік қосымшаның көмегімен Қазақстан Республикасының кез келген азаматы ҚР аумағында тіркелген дәрілік заттардың барлық тізбесін көре алады. Пайдаланушыға медицинада қолданылуы жөніндегі нұсқаулық, дәрі-дәрмектің суреті, қолдану тәсілдері, сертификаттар, дәрінің толық атауы, өндіруші туралы ақпарат, тіркеу куәлігінің нөмірі, баламалар тізбесі, шекті бағалар және тағы басқалар қол жетімді.

Артықшылықтары:

- қосымша тегін жүктеледі және пайдаланылады;
- интуитивті және ыңғайлы интерфейс;
- интерфейс орыс, қазақ және ағылшын тілдерін қолдайды.

Кемшіліктері:

- жұмыс істемейтін функционал;
- дәрі-дәрмектің бағасы жайлы ақпарат нақтылыққа сәйкес келмейді;
- дәрілердің бағасы көп жерде көрсетілмеген;
- қосымшадағы кірістірілген сканер жұмыс істемейді. [9]

## 2 Технологиялық бөлімі

### 2.1 Қолданылған құралдарға шолу

#### 2.1.1 Flutter SDK

Flutter – Google компаниясы ұсынған, бір кодты бірнеше рет қолдану мүмкіндігі бар кроссплатформалық қолданбаларды құруға арналған фреймворк болып табылады. Платформаның ауқымы кең, оған веб-қосымшалар, Android және iOS негізінде әзірленген мобильді қосымшалары, Windows, MacOS, Linux операциялық жүйелері негізінде құрылған жұмыс үстеліне арналған графикалық қосымшалар, және т.б. [1]

Flutter-мен жұмыс істеудің ерекшелігі – әр түрлі платформаларға арналған қосымшалар бірдей кодқа ие бола алады. Қолданылатын платформалар эквивалент емес болғандықтан, кодтың кейбір жеке бөліктері белгілі бір ОЖ үшін, мысалы, iOS үшін конфигурациялануы керек, бірақ соған қарамастан кодтың көп бөлігі бірдей болуы мүмкін. Бұл әзірлеушілерге барлық қолдау көрсетілетін платформалар үшін қосымшалар жасауға уақыт пен ресурстарды айтарлықтай үнемдеуге мүмкіндік береді.

Flutter негізгі үш компоненттен тұрады (2.1-сурет):

*Embedder.* Соңғы платформада өзара әрекеттесуге және іске қосуға жауап беретін платформаға тәуелді код.

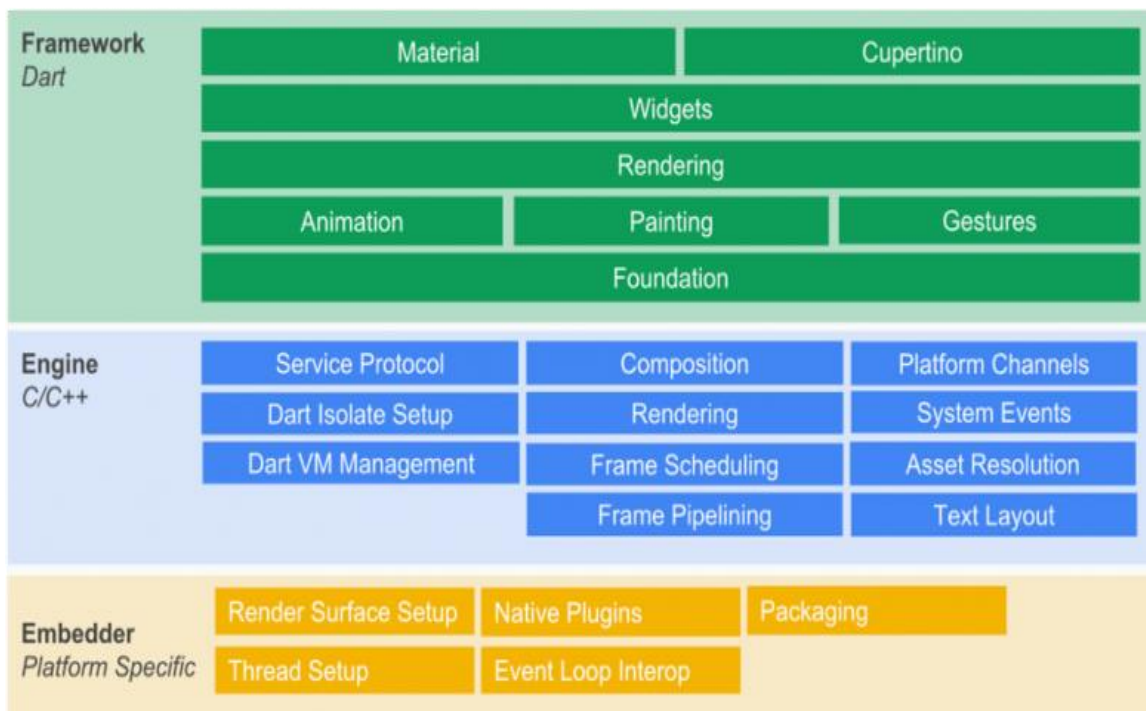
*Engine.* C / C++ тілінде жазылған. CPU-мен, Skia-2D сурет кітапханасы және Android-тан алынған мәтінді көрсетуге арналған ішкі жүйемен жұмыс жасау үшін Dart-ты абстракция деңгейі ретінде қамтиды.

*Framework.* Толығымен Dart тілінде жазылған ең үлкен бөлігі. Қабаттарға бөлу кез-келген күрделі қосымшаларды құруды жеңілдетеді. Material-UI-ден дайын виджеттерді немесе негізгі виджеттер негізінде спецификалық компоненттерді іске асыру арқылы пайдалануға болады. Анимациямен және қимылдарды өңдеумен икемді жұмыс істеу мүмкіндігі қарастырылған. Осылайша, жылдам жұмысты жоғары деңгейлі виджеттермен біріктіруге болады. [7]

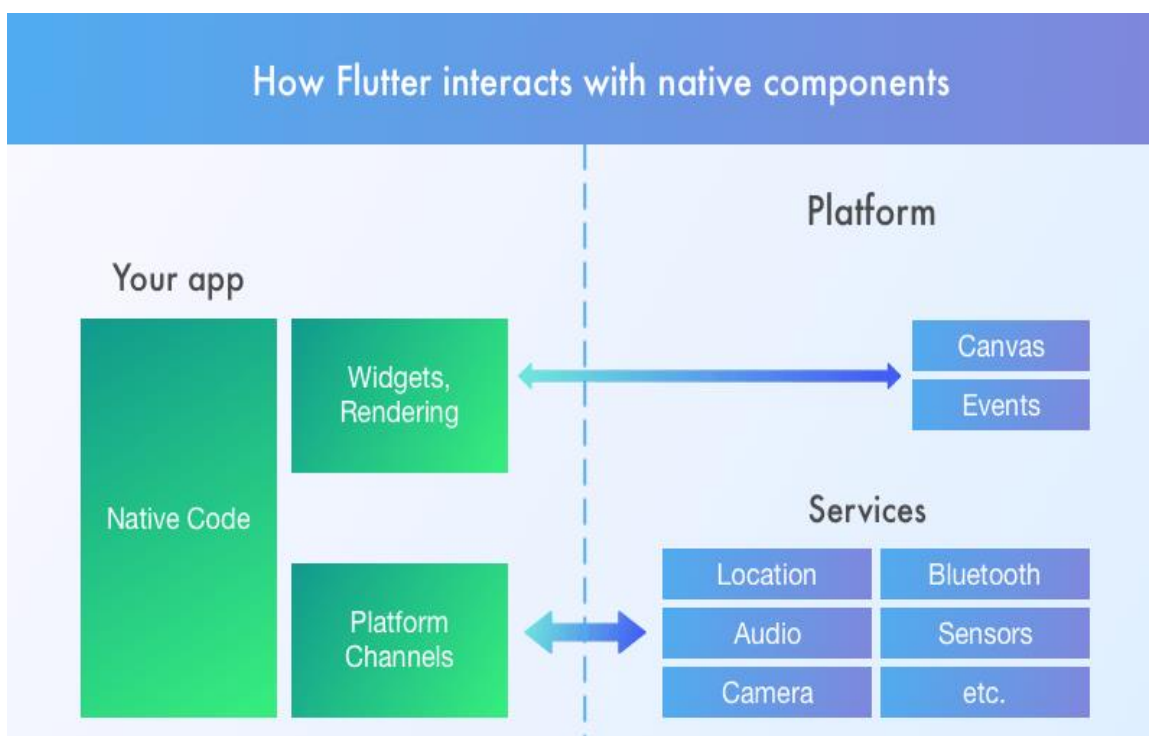
Flutter архитектурасының ерекшелігі – ол әр пиксельді өзі салады, қимылдар мен анимацияларды басқарады. Ол React Native сияқты OEM виджеттерін пайдаланбайды. Оның орнына Flutter негізгі мобильді платформаларға арналған екі виджет жиынтығын жасады: Android үшін Material және iOS үшін Cupertino. Осылайша олар екі мобильді платформадың мінез-құлқын толығымен қайталай отырып барлық UI компоненттердің суретін қайта салды. Flutter архитектурасы 2.2-суретте көрсетілген.

Әзірлеуші (разработка) тілі ретінде Dart бағдарламалау тілі қолданылады. Қосымшаны құру кезінде Flutter кодын Android, iOS немесе басқа платформада іске қосуға болатын Dart AOT (қосымшаны іске қосар алдында компиляциялау) көмегімен жергілікті қолданба кодына аударады. Алайда, оны жеделдету үшін

қосымшаны әзірлеу кезінде Flutter JIT (қосымшаны іске қосу кезінде жинақтау) қолданады. [3]



2.1-сурет – Flutter компоненттері



2.2-сурет – Flutter архитектурасы

Flutter – бұл салыстырмалы түрде жаңа құрылым. Фреймворктың прототипі 2015 жылы пайда болғанымен және алғашқы альфа нұсқасы 2017

жылдың мамырында шығарыған болатын, алайда алғашқы тұрақты шығарылымы – Flutter 1.0 тек 2018 жылдың желтоқсанында шығарылды. Соған қарамастан, қысқа уақыт аралығында жаңа функционалдылықты қамтитын және бар қателері түзетілген фреймворктың түп нұсқалары үнемі шығарылып, жаңартылып отырды. Осы жарияланым кезінде ең соңғы тұрақты шығарылым – 2021 жылы наурызда шыққан Flutter 2.0.

Бағдарлама кодын жазу үшін кез-келген мәтіндік редакторды қолдануға болады, содан кейін қосымшаны құрастыру үшін Flutter SDK командалық жолының утилиталарын қолдана аласыз. Алайда, Android Studio және IntelliJ IDEA сияқты орталар үшін, сондай-ақ Visual Studio Code мәтіндік редакторы үшін Google разработканы жеңілдететін плагиндерді шығарды. Сондықтан Flutter үшін Android Studio және IntelliJ IDEA, Visual Studio Code жиі қолданылады. [2]

### **2.1.2 Dart бағдарламалау тілі**

Dart – бұл негізінен Google жасаған және әлі де дамып келе жатқан жоғарғы деңгейлі бағдарламалау тілі. Дартты ЕКӨК (Еуропалық компьютер өндірушілер қауымдастығы) стандарттаған. Қазіргі уақытта жалғыз JavaScript-ке балама бағдарламалау тілі болып табылады. Бірінші нұсқасы 2011 жылы қол жетімді болды. [6]

Тілді жасаушы Марк Миллер Dart-тың жасалу себебі, «JavaScript-те «архитектуралық кемшіліктер» бар, оларды түзетуге болмайды» деп түсіндірді. Dart-та жоғарыда аталған кемшіліктер жоқ.

Бағдарламалау тілін әзірлеу кезінде оны жасаушылар:

- Dart тілін тез меңгеру мақсатында, кіру шегін төмендетуге;
- жылдам және ыңғайлы әзірлеу үшін құрылымдауға;
- бағдарламалау тілінде жазылған қосымшалар браузерлердегідей, және т.с.с. орталарда (мысалы, смартфондарда) жылдам жұмыс жасау үшін, өнімділігін арттыруға тырысты.

Dart-тың бірнеше артықшылығы бар:

- тілді меңгерудің қарапайымдылығы;
- кез келген JavaScript бағдарламашы Dart кодының синтаксисін тез игере

алады.

Тек осы бағдарламалау тілінің негізгі қағидаларын білу қажет:

- Жоғары өнімділік. Dart-та жасалған бағдарламалар, көп жағдайда javascript-те жасалған бағдарламаларға қарағанда, жылдамырақ жұмыс істейді;
- Қатаң бір типке келтіру. Java Script тіліндегі бағдарламаларға қарағанда, Dart тілінде кодты жазу кезінде қателіктер жіберу айтарлықтай қиын.

– Dart бағдарламашыларының жоғары жалақысы. Бұл бағдарламалау тілі, мысалы, Python сияқты кең таралмаған. Алайда Dart тілінде жаза алатындар нарықта әлдеқайда жоғары бағаланады. Сонымен қатар, тіл белсенді дамып, жыл сайын танымал болып келеді;



– Құжаттама. Dart-қа Google аудармашы (интерпретатор) әзірлеп жатқандықтан, тілдің барлық функциялары егжей-тегжейлі жазылған. Бұл оқыту және әзірлеу (разработка) кезінде туындайтын сұрақтарға жедел жауап алуға мүмкіндік береді. [3]

Dart бағдарламалау тілінің басқа бағдарламалау тілдерінен басты айырмашылықтарының бірі – адам тіліне ұқсас синтаксиспен оқудың жеңілдеудігінде. Мұнда командалар азырақ, бірақ көп параметрлер қолданылады. Сіз өзіңіздің айнымалыларыңыздың аттарын таңдай аласыз, сонда сіз жазған кез-келген код түсінікті болады.

### **2.1.3 IDE Android Studio**

Android Studio – бұл Android қосымшаларын жасауға арналған ресми интеграцияланған даму ортасы (IDE). Оның негізі IntelliJ IDEA, Java-да бағдарламалық қамтамасыздандыруды дамытатын орта және кодты редакциялау құралдары мен өңдеу құралдарын қамтиды. [5]

Android операциялық жүйесінде қосымшаларды әзірлеуге қолдау көрсету үшін Android Studio бағдарламасы Gradle негізінде құрастыру жүйесін, эмуляторды, шаблондарды қолданады және Github-пен интеграцияланады.

Android Studio-дағы әрбір жобада бастапқы кодпен және ресурстық файлдармен бір немесе бірнеше модаль бар. Бұл модальдарға Android қосымшалары, кітапхана модульдері және Google AppEngine модульдері кіреді.

Android Studio жұмыс істеп тұрған бағдарламаға кодты және ресурстардағы өзгерістерді жіберу үшін Instant Push қолданады. Код редакторы әзірлеушіге код жазуға көмектеседі және кодтау бойынша аяқтауды, рефракциялауды және талдауды ұсынады. Содан кейін Android Studio-да жасалған бағдарламалар APK форматқа сәйкес Google Play Store-ға жіберу үшін жинақталады.

Бағдарламалық жасақтама 2013 жылдың мамырында алғаш рет Google I/O-де жарияланды, ал алғашқы тұрақты нұсқасы 2014 жылдың желтоқсанында шығарылды. Android Studio Mac, Windows және Linux жұмыс үстелі платформаларында қол жетімді. Ол Eclipse ADT Android үшін қосымшаларды әзірлеуге арналған негізгі құрылым ретінде ауыстырылды.

### **2.1.4 AVD Manager**

Android Studio-ның бәсекелестеріне қарағанда басты артықшылықтарының бірі пакетке салынған Android құрылғыларының эмуляторының болуы, оған әзірлеу кезеңінде дайын жобаларды тестілеуге, орналасқан утилиталардың дұрыстығын тексеруге болады. Әр тестілеуге нақты құрылғыны қолданғаннан гөрі, бұл әлдеқайда ыңғайлы болып табылады.

Android Studio эмуляторы функционалдылыққа бай. Оның қосымшаны әртүрлі экран конфигурацияларда тексеру үшін, нақты арақатынасты, қажетті ажыратымдылықты имитациялауға, смартфонның, планшеттің немесе басқа Android құрылғысының техникалық сипаттамаларын тексеру үшін жеткілікті.

### 3 Жобалау бөлімі

#### 3.1 «Дәріханалар бойынша мобильдік қолданба-агрегатор» қосымшаны UML модельдеу тілі көмегімен жобалау

«Дәріханалар бойынша мобильдік қолданба-агрегатор» қосымшасын жобалау барысында UML – диаграммалар құрылды. Диаграммаларды құру үшін тегін шаблондары бар диаграммаларды құруға арналған [online.visual-paradigm.com](http://online.visual-paradigm.com) веб-сайты қолданылды. Диаграммаларды құрмас бұрын, жалпы UML дегеніміз не? Қандай түрлері бар? деген сұрақтарға жауап қарастырайық және қысқаша сипаттама беріп өтейік.

##### 3.1.1 UML біріңғай моделдеу тілі

UML (ағылш. Unified Modeling Language – модельдеудің унифицирленген тілі) – бизнес-процестерді модельдеу, жүйелік жобалау және ұйымдық құрылымдарды бейнелеу үшін бағдарламалық қамтамасыз етуді әзірлеу саласындағы объектілі модельдеу үшін графикалық сипаттама тілі. Бұл тіл кең профильге ие және жүйенің абстрактілі моделін құру үшін әр түрлі графикалық белгілерді қолданатын ашық стандарт болып табылады. [10]

UML бағдарламалық жасақтаманың барлық түрлерін анықтауға, көрнекілікке, құжаттамаға және дизайнға арналған. UML диаграммасының өзі бағдарламалау тілі емес екенін ескерген жөн, бірақ оның негізінде жеке кодты құру мүмкіндігі қарастырылған.

UML міндеттерінің бірі – команда ішінде және тапсырыс берушімен байланыс кезінде байланыс құралы ретінде қызмет ету. Диаграммаларды қолданудың мүмкін жағдайларын қарастырайық:

*Жобалау.* UML диаграммалары үлкен және кіші бөлшектерді жинауға және қосымшаның жақтауын (схемасын) салуға болатын үлкен жобалардың архитектурасын модельдеуге көмектеседі. Кейіннен код оған негізделетін болады.

*Кері инженерия* – қолданыстағы қолданбалы кодтан UML моделін құру, кері құрылыс. Оны, мысалы, код жазылған, бірақ құжаттама толық емес немесе жоқ қолдау жобаларында қолдануға болады.

Модельдерден мәтіндік ақпаратты шығарып алуға және салыстырмалы түрде оқылатын мәтіндер жасауға болады. Мәтін мен графика бірін-бірі толықтырады.

Барлық UML диаграммаларын екі топқа бөлуге болады, олардың біріншісі жалпы диаграммалар. Жалпы схемалар модельдеу тақырыбынан іс жүзінде тәуелсіз және кез-келген бағдарламалық жасақтамада субъектінің аймағына, шешім аймағына және т.б.

UML мына диаграммалардан тұрады:

- құрылымдық диаграммалар;
- мінез-құлық диаграммалар;
- өзара әрекеттесу диаграммалары.

### 3.1.2 Прецеденттер диаграммасы (Use Case diagram)

Бұл әртүрлі жүйелерді талдау үшін жиі қолданылатын UML мінез-құлық диаграммасының түрі. Олар жүйеде әртүрлі рөлдер типтерін және осы рөлдердің жүйемен өзара әрекеттесуін көзбен көруге мүмкіндік беретін диаграмма болып табылады. Диаграмманың негізгі мақсаты - тұтынушыға, соңғы пайдаланушыға және әзірлеушіге жобаланған немесе бар жүйені бірлесіп талқылауға мүмкіндік беріп, функционалдылық пен мінез-құлықты сипаттау. Диаграмманың негізгі элементтері - актер және прецедент (нұсқа).

Актер, қатысушы дегеніміз – пайдалану жағдайлары немесе объектілерімен (жүйемен, ішкі жүйемен немесе класспен) өзара әрекеттесу кезінде орындалатын логикалық байланысты рөлдердің жиынтығы.

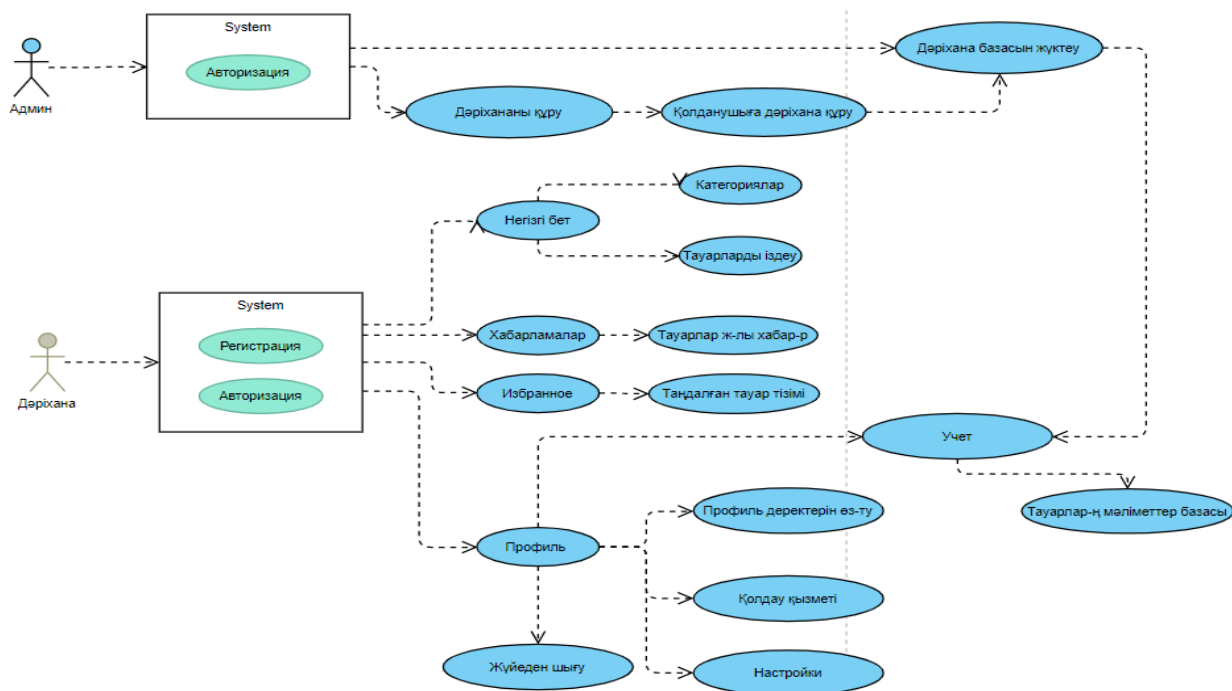
Қолдану жағдайы (прецедент) – бұл қатысушы байқайтын нәтижеге әкелетін жүйемен орындалатын дәйекті оқиғалар жиынтығының (нұсқаларын қоса) сипаттамасы.

Төмендегі 3.1-суретте қолданушыға, 3.2-суретте дәріханаға арналған прецедент диаграммасы көрсетілген.



3.1-сурет – Қолданушыларға арналған прецеденттер диаграммасы





**3.2-сурет – Деріханаға арналған прецеденттер диаграммасы**

### 3.1.3 Класстар диаграммасы (Class diagram)

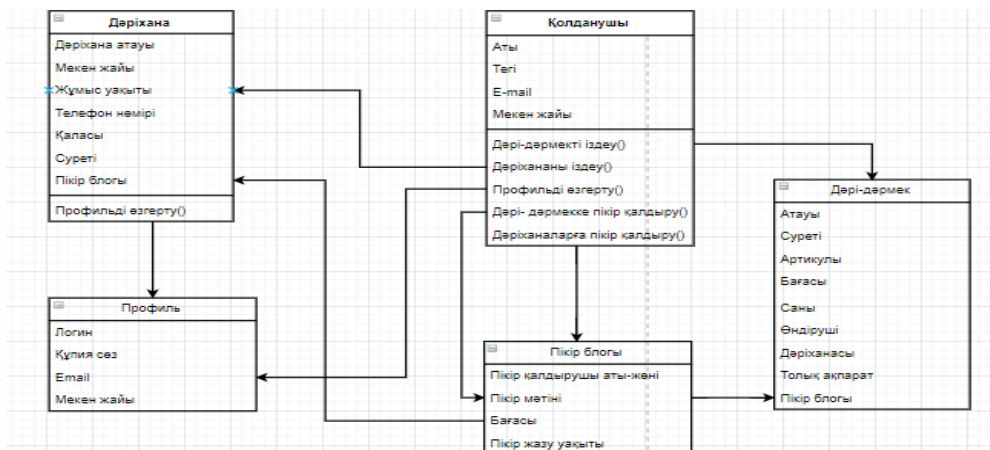
Класстар диаграммасы UML диаграмма редакторының негізгі элементі болып табылады, өйткені көбінесе қосымшалар класстар диаграммасынан құрылады. Класстар диаграммасы дегеніміз – статикалық, декларативті модель элементтерінің жиынтығы. Класс диаграммаларын тікелей жобалау кезінде де, яғни жаңа жүйені құру процесінде де, кері инженерияда – қолданыстағы және қолданылған жүйелерді сипаттауда да қолдануға болады. Класс диаграммасындағы ақпарат қолданбаның бастапқы кодына тікелей бейнеленеді. Осылайша, класс диаграммасы жобалаудың соңғы нәтижесі және даму процесінің бастапқы нүктесі болып табылады.

Арнайы таңбаларды қолдана отырып, класс диаграммасында бағдарлама деректерінің түрлері және олардың арасындағы қатынастар көрсетілген, дегенмен кейбір жағдайларда басқа элементтер – пакеттер және тіпті класс даналары да қолданыла алады.

Класстар диаграммасының артықшылықтары:

- класс диаграммасын өте күрделі ақпараттық жүйелер үшін де деректер модельдерін иллюстрациялайды;
- бұл нақты кодты зерттемес бұрын қосымшаның құрылымы туралы жалпы ақпарат береді;
- бұл қосымшаның жалпы схемаларын жақсы түсінуге көмектеседі;
- программалауға қажетті кодты бөліп көрсететін егжей-тегжейлі сызбалар жасауға мүмкіндік береді;
- әзірлеушілер мен басқа мүдделі тараптар үшін пайдалы.

Жүйенің класстар диаграммасын төмендегі 3.3-суреттен бақылауға болады.

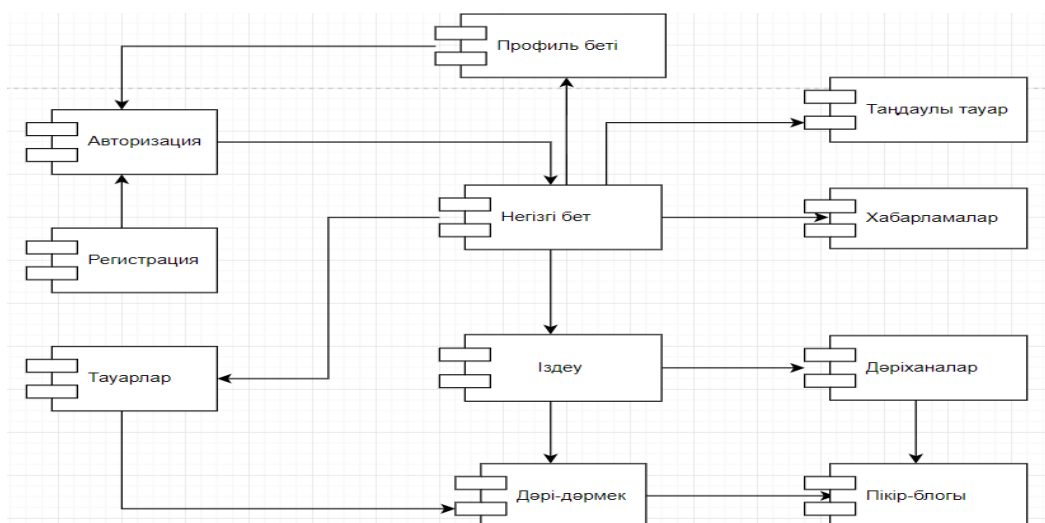


3.3-сурет – Жүйенің класстар диаграммасы

### 3.1.4 Компоненттер диаграммасы (Component diagram)

UML компоненттерінің диаграммалары объектілік-бағытталған жүйелердің физикалық жақтарын модельдеу үшін қолданылады, олар компоненттер жүйелерін визуалдау, анықтау және құжаттау үшін және алға және кері инженерия арқылы орындалатын жүйелерді құру үшін қолданылады. Компоненттік диаграммалар дегеніміз жүйенің статикалық көрінісін модельдеу үшін жиі қолданылатын жүйенің компоненттеріне бағытталған класстық диаграммалар.

Компоненттік диаграмманың мақсаты – жүйенің әртүрлі компоненттері арасындағы байланысты көрсету. Дәрiхана жүйесінің басты компоненттерінің диаграммасы төменде 3.4-суретте көрсетілді.

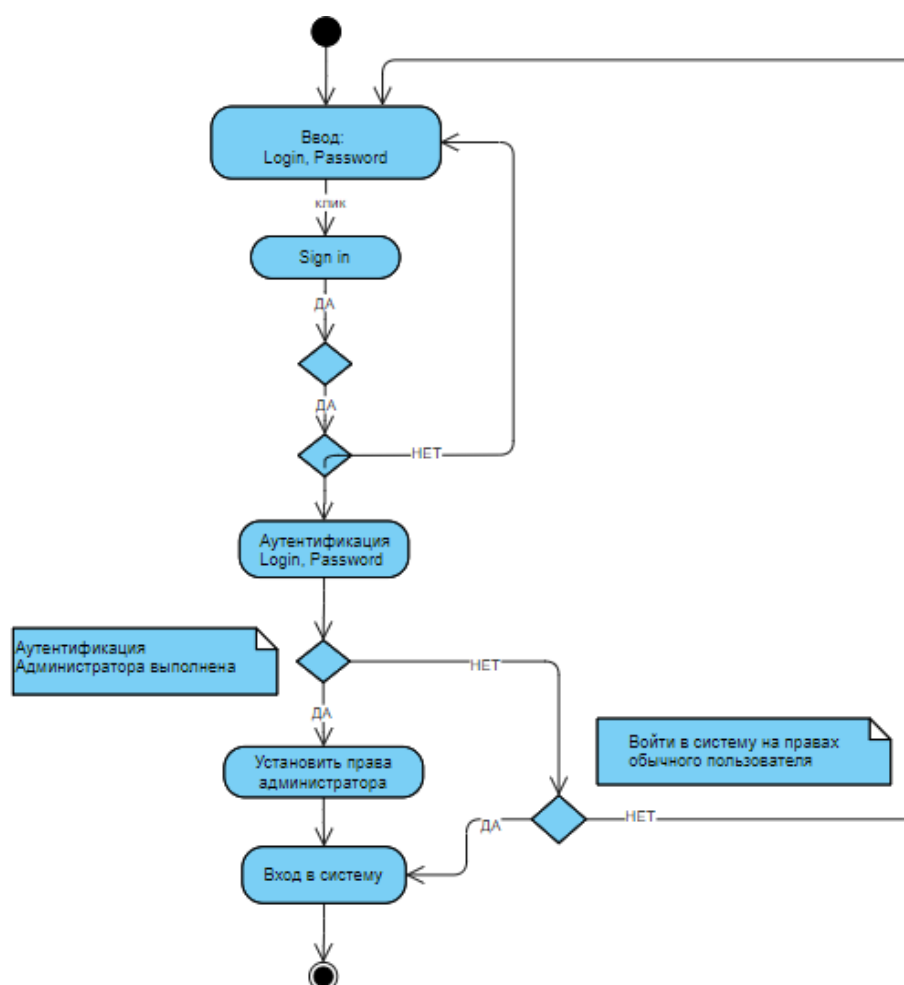


3.4-сурет – Жүйенің компоненттер диаграммасы

### 3.1.5 Белсенділік диаграммасы (Activity diagram)

Белсенділік диаграммасы – күй диаграммасында сипатталған әрекеттерді көрсететін UML диаграммасы. Белсенділік деп бағынатын элементтердің үйлесімді дәйекті және параллель орындалуы түріндегі орындалатын мінез-құлықты спецификациялау – кірістірілген әрекеттер мен жеке әрекеттер деп түсінеміз. бір түйіннің шығуларынан екінші түйіннің кірістеріне өтетін ағындармен өзара байланысты әрекет.

Белсенділік диаграммалары бизнес-процестерді, технологиялық процестерді, дәйекті және параллельді есептеуді модельдеуде қолданылады.

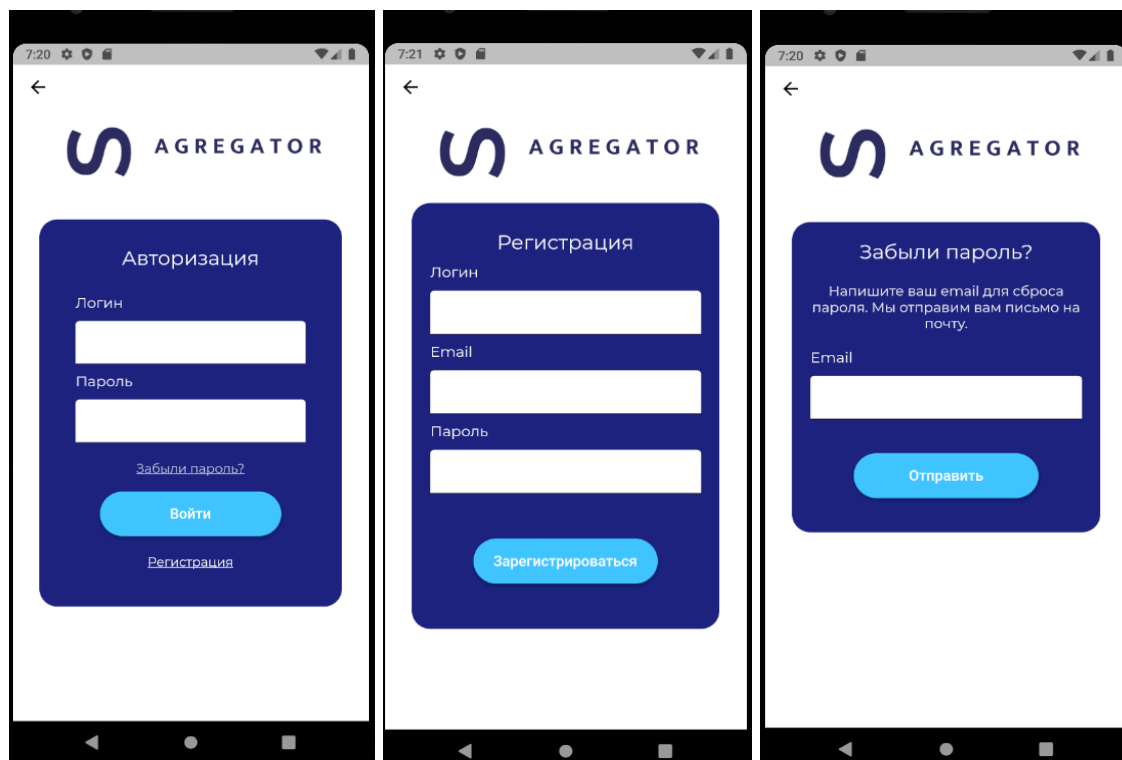


3.5-сурет – Жүйенің белсенділік диаграммасы

### 3.2 Мобильдік қолданба-агрегаторының қолданушылар интерфейсін әзірлеу

Мобильді қосымшаның бірінші бетінде авторизация беті ашылады. Төмендегі 4.1-суретте көрсетілгендей авторизация бетінде қосымшаға тіркелуге

болады. Тіркелу бетінде логин, email және пароль беру арқылы жеке профильді құрып аламыз, сол арқылы қосымшаға тіркелеміз. Құпия сөзді ұмытып қалған жағдайда, құпия сөзді почтаға арқылы қалпына келтіруге болады.



#### 4.1-сурет—Авторизация, тіркелу, құпия сөзді қалпына келтіру беті

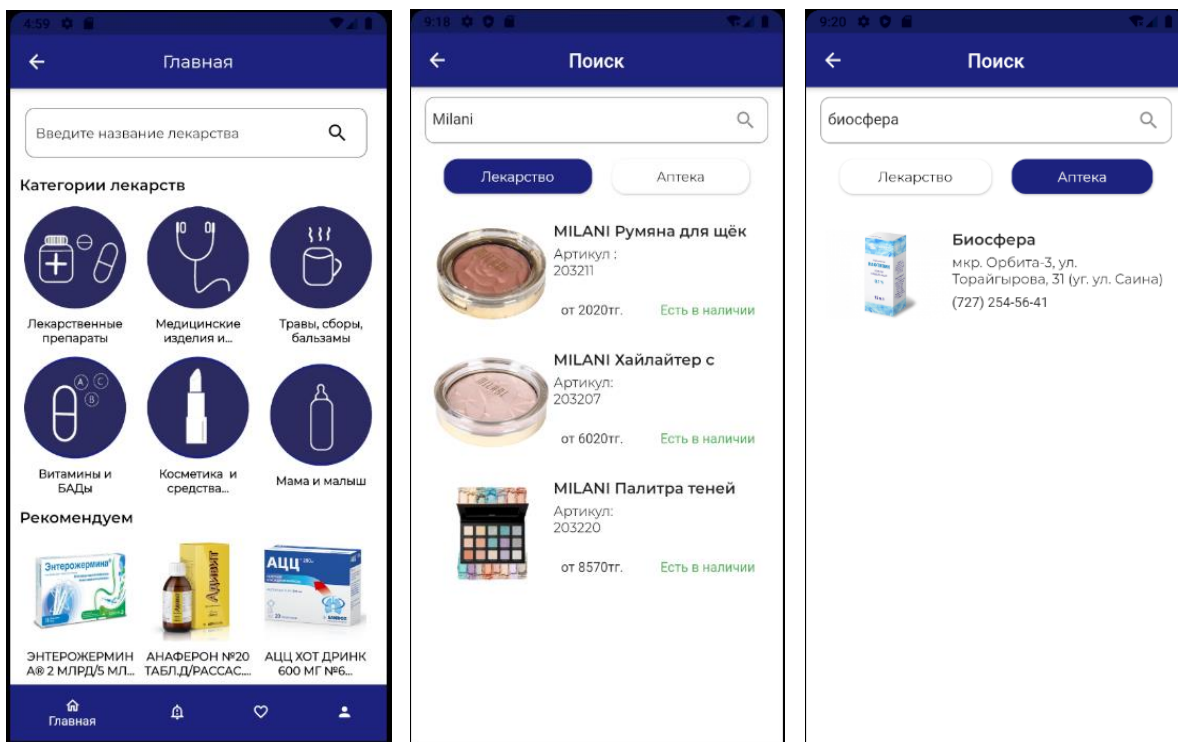
4.2-суретте қосымшаға авторизация жасағаннан кейінгі негізгі бет, тауарды іздеу беті көрсетілген. Негізгі бетте әрбір дәріханаларды жалпы тауарларына сәйкес категорияларға бөлінген, және де пайдаланушыға арналған «ұсыныс» бөлімі арқылы ең тиімді ұсынысты таңдау жасауға мүмкіншілік бар. Агрегатордың арнайы іздеу бетінде дәрі-дәрмектің немесе дәріхананың атауын еңгізу арқылы қолданушы қалаған тауарды, дәріханаларды тауып алу функциясы қарастырылған. Іздеу беті тауардың суреті, толық атауы, артикул, бағасы және де қол жетімділігі туралы ақпаратты қамтиды,

Кез-келген категория ішіне кіретін болсақ, төменде 4.3-суретте көрсетілгендей сол категорияға сәйкес дәрі-дәрмектің немесе басқа да тауарлардың тізімін көре аласыз. Дәрі-дәрмектің үстінен басу арқылы, толық ақпаратқа қол жеткізе аласыз. «Наличие» бөліміне өту арқылы тауардың саны, бағасы, қандай дәріханада қол жетімді екенін бақылай аласыз, және де бағасын салыстыра отырып, тиімді ұсынысты таңдау мүмкіндігі бар.

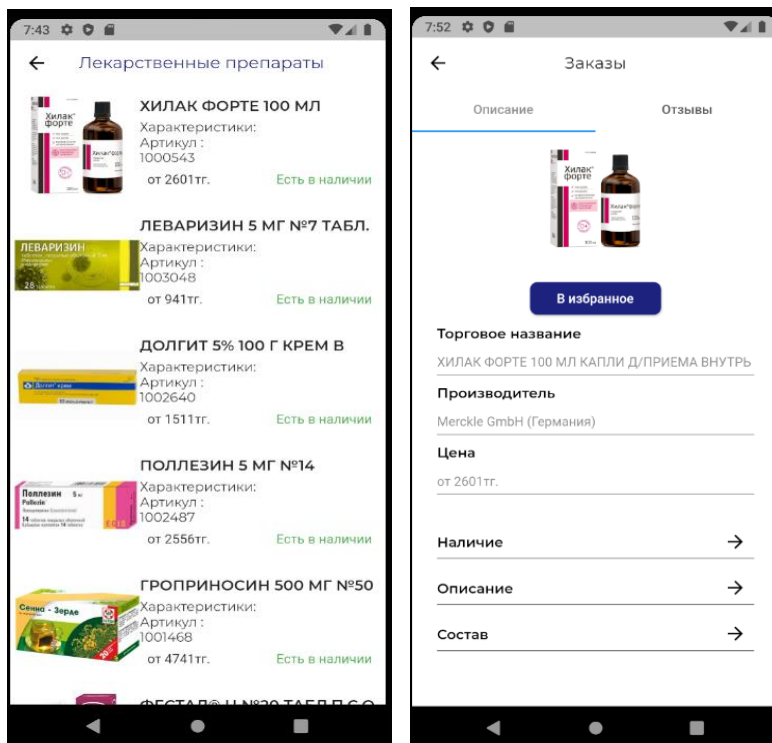
Сондай-ақ, 4.4-сурет көріп тұрғаныңыздай, «Отзывы» бөліміне өту арқылы дәрі-дәрмекті қолданған адамдардың дәрі туралы қалдырып кеткен пікірлерін оқып, дәрі-дәрмектің, басқа да тауарлардың көмегі, әсері бар немесе жоқтығына көз жеткізуге болады. «Оставить отзыв» тетігі арқылы тауар жайлы ойыңызбен бөлісе алу мүмкіндігі қарастырылған. Пікір қалдыру үшін қолданушы аты-



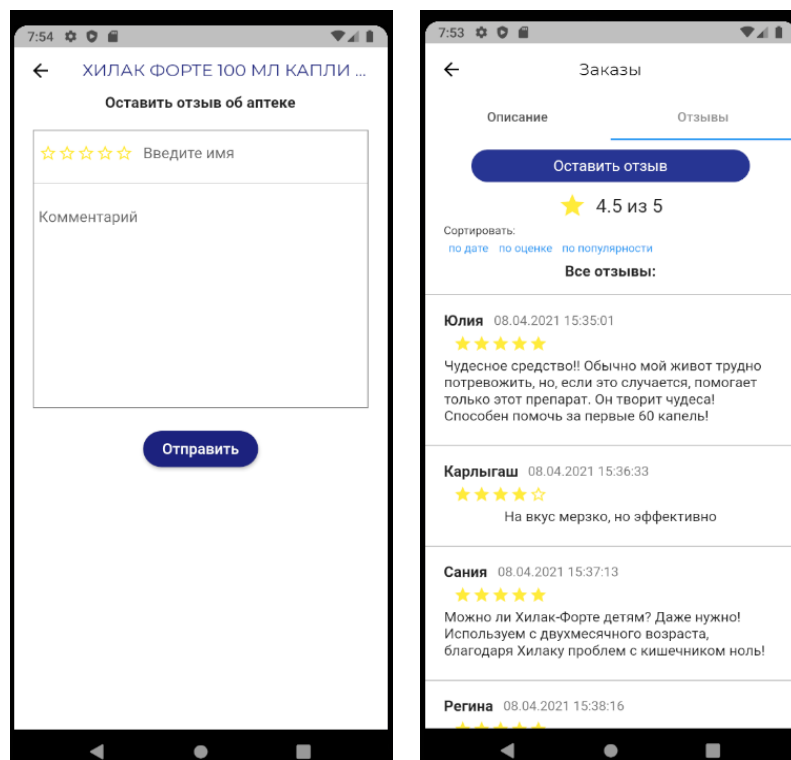
жөнүн, дәрі-дәрмекке қоятын бағасын, және пікір жазу формасын толтыруы қажет.



4.2-сурет – Негізгі бет, тауарды іздеу беті



4.3-сурет – Дәрілер тізімі беті, дәрі жайлы ақпарат беті

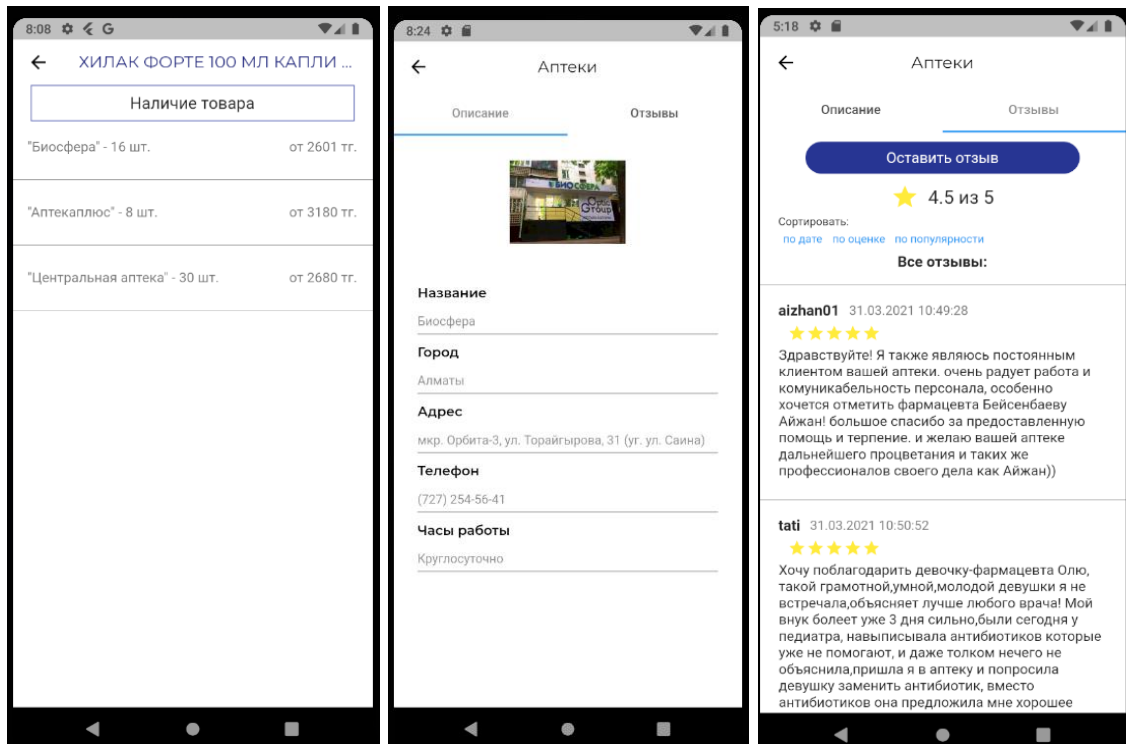


#### 4.4-сурет – Дәріхана жайлы ақпарат беті, дәрі-дәрмектің пікір беті

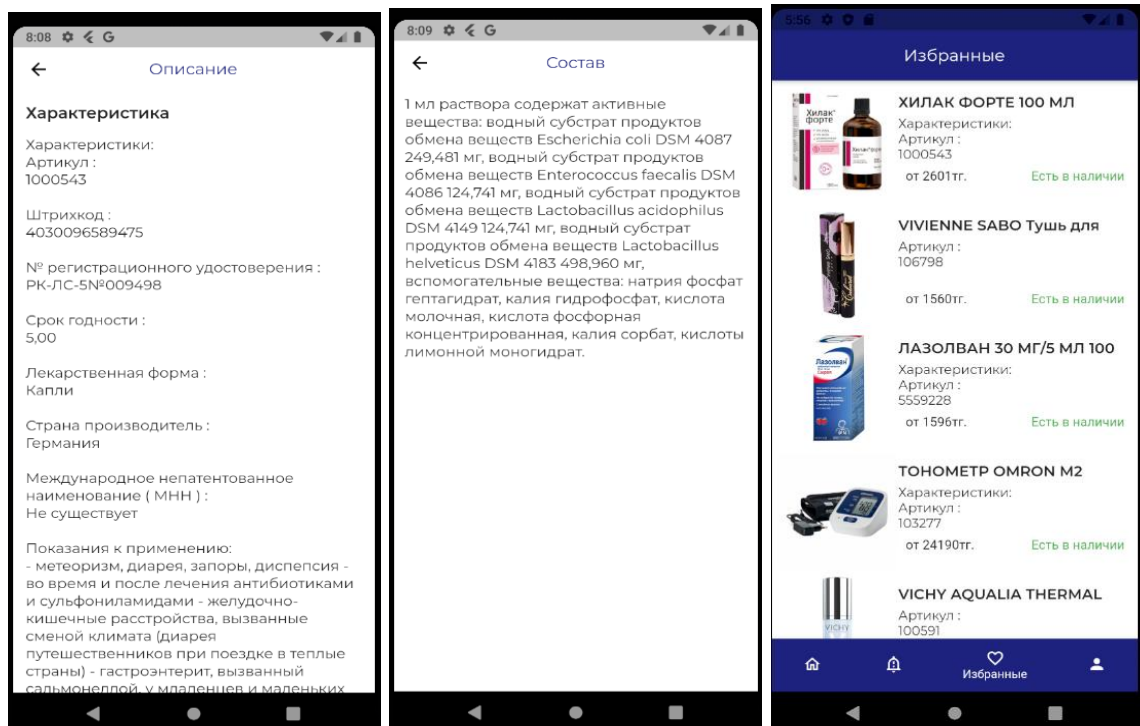
Сонымен қатар, 4.5-суретте көрсетілгендей дәріханалар тізімінен белгілі бір дәріхананың үстінен бассаңыз, дәріхана туралы толық ақпарат көрсетілген бетке қол жеткізе аласыз. «Отзывы» бөліміне өту арқылы дәріхана келушілерінің дәріхана туралы қалдырып кеткен пікірлерін оқып, дәріхананың қызметі жайлы пікір-сараптарын қарау мүмкіндігі бар. «Оставить отзывы» тетігі арқылы сіз дәріхана жайлы ойыңызбен бөлісу қарастырылған. Пікір қалдыру үшін қолданушы аты-жөнін, дәрі-дәрмекке қоятын бағасын, және пікір жазу формасын толтыруы қажет.

4.5-суретте дәрі-дәрмектің «Описание», «Состав» бөліміне өтсеңіз, дәрі жайлы толық ақпарат көрсетілген. Сондай-ақ, 4.3-суретте көрсетілген «В избранное» тетігі арқылы таңдаулы тауарлар бетіне қалаған тауарыңызды еңгізіп қою мүмкіндігі бар. «Избранное» беті 4.6-суретте көрсетілген.

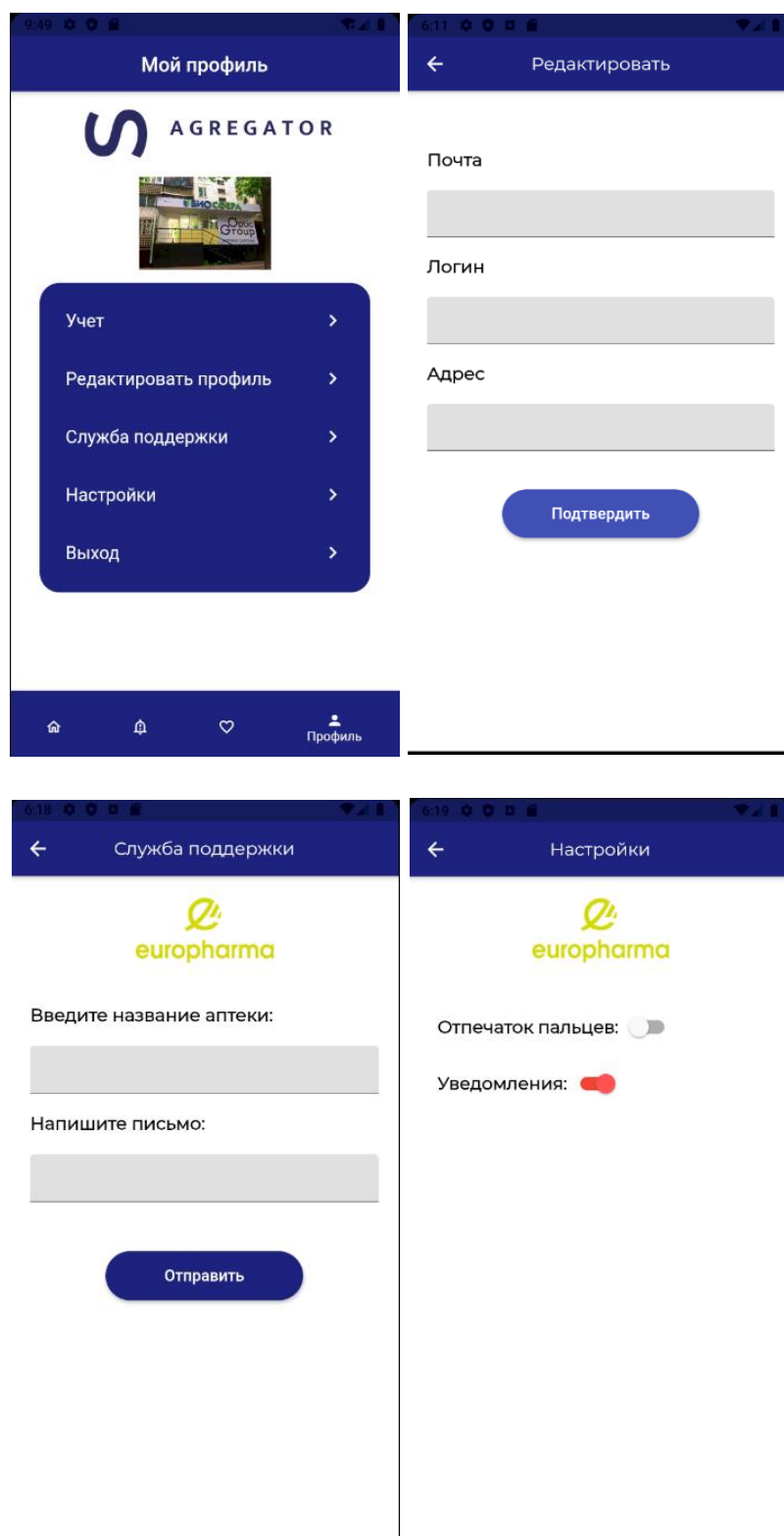
Төмендегі 4.7-суретте қолданушының жеке профиль беті көрсетілген. Жеке профиль бетінде өзіңіз жайлы ақпаратты өзгертуге болады. Қолданушы атында дәріхана болса, 4.9-суретте көрсетілгендей «Учет» беті ашылады, ал қолданушының атында тіркелген дәріхана жоқ болған жағдайда, «Учет» беті қосылмайтын болады. Дәріхананы <http://agregator-admin.herokuapp.com/login> админ панельде супер пользователь ғана дәріхана аккаунтын тіркеп бере алады. Дәріхана қолданушы атынан құрылса, <http://agregator-admin.herokuapp.com/login> арқылы, құрылған дәріхананың базасын уақытылы жаңартып отыра аламыз. Ол жерде дәріхананың мәліметтер базасын JSON форматындағы файлды жүктеу арқылы жаңартып отыра аламыз. «Служба поддержки» бетіне өтіп, дәріхананың қолдау көрсету қызметіне хат жазуға, сондай-ақ параметрлер бөлімінде сізге қажетті функционалды қосуға болады.



4.5-сурет – Дәріханалар тізімі, дәріхана туралы пікір беті



4.6-сурет – Дәрі туралы ақпарат беті, таңдаулы тауарлар беті

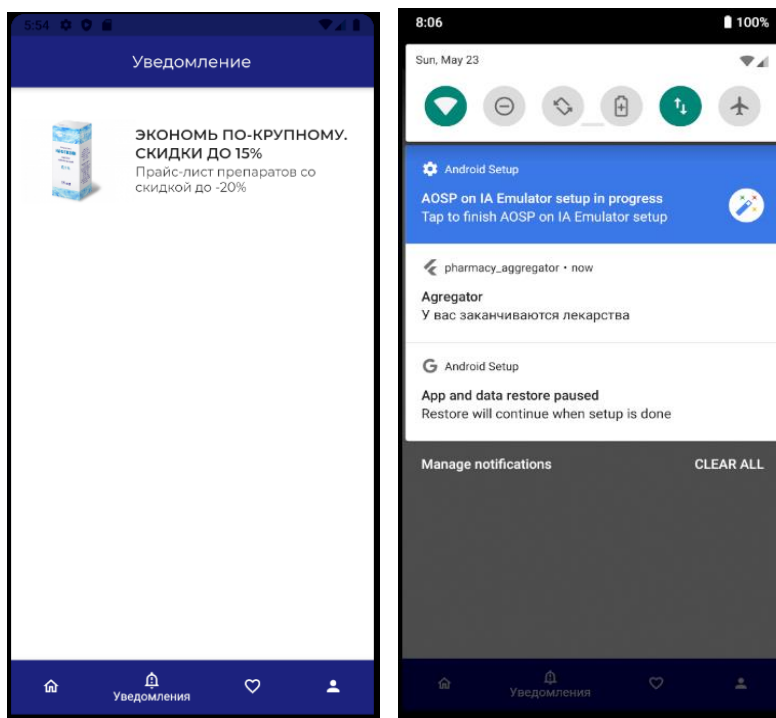


**4.7- сурет – Жеке профиль, жеке ақпаратты өзгерту беті, қолдау көрсету және параметрлер беті**

«Хабарламалар» бетінде қарапайым пайдаланушыларға әр дәріхананың тауарға берілген жеңілдіктері жайлы пуш-хабарламалардан тұратын тізім көрсетіледі (4.8-сурет). Ал дәріхана иелеріне арналған «Хабарламалар бөлімінде дәріхананың жеке профиліне тауарлардың (дәрі-дәрмектің, витаминдердің,

медициналық приборлардың, косметикалық өнімдердің, т.с.с) белгіленген көлемінен саны аз болса, тауардың бітуі жайлы хабарлама алып отырады. Сондай-ақ, белгілі бір тауарлардың жарамдылық мерзімі жайлы ақпаратты хабарлама арқылы алуға мүмкіншілік жасалған (4.6-сурет).

4.9-суретте дәріхана иелеріне арналған «Учет» беті көрсетілген. Онда дәріханалардағы қол жетімді дәрілердің атауы, саны, артикул, бағасы жайлы ақпарат сақталады. Бұл ақпарат арқылы, дәріхана иесі тауардың санын, орнатылған бағасын бақылып отыра алады.



4.8-сурет – Хабарламалар беті

id	Наименование лекарств	Код (Арт.)
1	MILANI Хайлайтер с галаграфическим эффектом luster light 03	2
2	DUCRAY KERTYOL PSO Бальзам увлажняющий 200мл	1
3	MILANI Румяна для щёк мини версия ROMANTIC ROSE 901	3
4	MILANI Палитра теней GILDED NUDE B9198A	5
5	VIVIENNE SAVO Тушь для ресниц Artistic 01 D215003101	7
6	AVENE CLEANANCE WOMAN Сыворотка 30 мл	4
7	YOKO SUN Салфетки влажные детские 0+ 100шт	28
8	YOKO SUN PREMIUM Подгузники-трусики XL 12-20кг 38шт	30
9	ИНФЛЮЦИД №60 ТАБЛ.	33
10	ЭССЕНЦИАЛЕ® Н №30 КАПС.ФОРТЕ	34
11	БАНЬКА АГАФЬИ Маска для лица Омолаживающая 100мл.	9
12	ТЕРМОМЕТР бесконтактный RAK-F103	22
13	ГРОПРИНОСИН 500 МГ №50 ТАБЛ.	35

id	Наименование лекарств	Код (Артикул)	Кол-во наличия	Цена
1	MILANI Хайлайтер с галаграфическим эффектом luster light 03	2	3	6020 тг
2	DUCRAY KERTYOL PSO Бальзам увлажняющий 200мл	1	11	7700 тг
3	MILANI Румяна для щёк мини версия ROMANTIC ROSE 901	3	2	2020 тг
4	MILANI Палитра теней GILDED NUDE B9198A	5	1	8570 тг
5	VIVIENNE SABO Тушь для ресниц Artistique 01 D215003101	7	26	1560 тг

**4.9-сурет – Дәріханаға арналған «Учет» беті**



## ҚОРЫТЫНДЫ

Дипломдық жобаны қорытындылай келе, нәтижесінде аптеканың Android, iOS платформасында жасалған мобильді қосымшасы жасалынды.

Жұмыстың басты мақсаты дәріхананың қолданба-агрегаторын құру және жүйені қолданушылардың уақытын үнемдеу, тұтынушылардың қалауларын қанағаттандыратындай қызмет көрсететін кроссплатформалық мобильді қосымша құрылды.

Онлайн дәріханалардың мобильдік қолданба агрегаторы тұтынушыға қажетті дәрі-дәрмектің тез әрі, дәл қол жетімділігін тексеру, сонымен қатар, шолуларды, жеңілдіктер мен артықшылықтарды салыстыру арқылы қолайлы ұсынысты таңдау мүмкіндігін береді. Сондай-ақ, қосымшада дәріхананың суреті, мекен-жайы, телефон нөмірі, жұмыс режимі туралы толық ақпарат қамтылғандықтан, дәріханалармен байланыс орнату қарастырылған.

Дипломдық жұмыстың нәтижесінде келесі негізгі тапсырмалар орындалды:

- мобильдік қосымшаның құрылымы құрастырылды;
- бағдарламалау тілі таңдалды;
- мәліметтер қоры таңдалды;
- тиімді және түсінікті программалық UI құрылды.

Жасалынған дипломдық жұмыстың нәтижесі болашақта көп адамдарған көмегі тиетініне сенімдімін. Көп қиындықтардың шешімін осы мобильді қосымша арқылы шешуге болады деген ойдамын.

Мен бұл дипломдық жобаның нәтижесі болашақта көптеген адамдарға көмектесе алатынына сенімдімін. Осы мобильді қосымшаның көмегімен көптеген мәселелердің шешімін табуға болады деп санаймын.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Flutter SDK кіріспе// Электрондық нұсқа <https://metanit.com/dart/flutter>.
- 2 Flutter қысқаша нұсқаулық// Электрондық нұсқа [https://ru.it-brain.online/tutorial/dart\\_programming/dart\\_programming\\_quick\\_guide/](https://ru.it-brain.online/tutorial/dart_programming/dart_programming_quick_guide/).
- 3 Dart – Structured web programming// Электрондық нұсқа <https://habr.com/ru/post/130064/>.
- 4 Dart тілінің ерекшеліктері мен артықшылықтары // Электрондық нұсқа <https://appfox.ru/blog/kod/preimushestvah-yazyka-programmirovaniya-dart>.
- 5 Android Studio бағдарламалау ортасы // Электрондық нұсқа <https://androidstudio.ru/>.
- 6 Dart language overview // Электрондық нұсқа <https://dart.dev/guides/language>.
- 7 Использование Flutter для разработки web-приложений, Касьянов А.С., Крючков В.С., Лукашевич С.А // Электрондық кітап.
- 8 I-teka // Электрондық нұсқа <https://iodroid.net/i-teka>.
- 9 Dari kz // Электрондық нұсқа <https://iodroid.net/darikz>.
- 10 UML диаграммы. Введение // Электрондық нұсқа <https://habr.com/ru/post/508710/>.

## **А Қосымшасы** (міндетті)

### **Техникалық тапсырма**

#### **А.1.1 Мобильдік қосымшаны әзірлеуге арналған техникалық тапсырма**

Техникалық тапсырма дәріханалардың мобильдік қолданба-агрегаторын әзірлеу мақсатында қолданылады. Бұл қосымшаны дәріханалар, қарапайым тұтынушылар, фармацевтикалық компаниялар пайдалана алады. Қосымша фармацевтика саласында қызмет көрсетуге бағытталған.

##### **А.1.1.1 Мақсаты**

«Дәріханалар бойынша мобильдік қолданба агрегаторы» қолданба қолданушылары мен дәріхана иелеріне көмектесу, дәріханалардың жұмысын жеңілдету үшін арналған.

##### **А.1.1.2 Қосымшаның сыртқы түріне қойылатын талаптар**

Қосымшаның сыртқы түрі интуитивті түсінікті және пайдалануға ыңғайлы болуы керек.

##### **А.1.1.3 Функционалдық сипаттамаларына қойылатын талаптар**

Пайдаланушыларға:

- пайдаланушыларды авторизациялау және тіркелу;
- өзіңіз туралы ақпаратты қосу және өңдеу;
- дәріханалардағы дәрі-дәрмектің қол жетімділігін тез тексеру;
- шолуларды, жеңілдіктер мен артықшылықтарды салыстыру арқылы қолайлы ұсынысты таңдау;
- дәріхананың суреті, мекен-жайы, телефон нөмірі, жұмыс режимі туралы толық ақпарат алу;
- жеңілдіктер туралы push хабарламалар алу.

Дәріхана иелеріне:

- пайдаланушыларды авторизациялау және тіркелу;
- өзіңіз туралы ақпаратты қосу және өңдеу;
- дәріханалардағы дәрі-дәрмектің қол жетімділігін тез тексеру;
- ауардың сақталу мерзімі, тауардың дәріханалардағы азайып баратқандығы жайлы push-хабарламалар алу.

#### **А.1.1.4 Сенімділікке қойылатын талаптар**

Енгізілетін ақпаратты бақылауды қамтамасыз ету. Деректер базасының қауіпсіздігін, сондай-ақ жұмыс істеу дұрыстығын алдын ала қарастыру.

#### **А.1.1.5 Құрамы мен техникалық құралдар параметрлеріне қойылатын талаптар**

Қосымша Android, iOS операциялық жүйелерінде жұмыс істеуі керек. Мобильді қосымшаны әзірлеу үшін келесі техникалық талаптарды анықтау қажет:

- Flutter SDK;
- Dart;
- PostgreSQL;
- Django.

#### **А.1.1.6 Бағдарламалық құжаттамаға қойылатын талаптар**

Жазылған бағдарламалық код бойынша түсініктеме әзірлеушінің қалауы бойынша берілуі керек.

## Б Қосымшасы (міндетті)

### 1. main.dart

```
import 'dart:io';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pharmacy_aggregator/core/constants.dart';
import 'package:pharmacy_aggregator/pages/authorization/sign_in_page.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'bottom_navigation_page.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  final FirebaseMessaging _firebaseMessaging = FirebaseMessaging();
  FlutterLocalNotificationsPlugin localNotificationsPlugin;
  AndroidInitializationSettings androidInitializationSettings = new
  AndroidInitializationSettings('@mipmap/ic_launcher');
  IOSInitializationSettings iosInitializationSettings = new IOSInitializationSettings();
  final GlobalKey<ScaffoldState>();
  Future<bool> isRegister() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    var isReg = prefs.getBool('isReg');
    return isReg;
  }
  var signInPage = SignInPage();
  var navPage = BottomNavigationPage();
  @override
  Widget build(BuildContext context) {
    InitializationSettings initializationSettings = new InitializationSettings(
      android: androidInitializationSettings, iOS: iosInitializationSettings );
    localNotificationsPlugin = FlutterLocalNotificationsPlugin();
    localNotificationsPlugin.initialize(initializationSettings);
    if (Platform.isIOS) iOS_Permission();
    _configureFirebaseListeners(context);
    _getToken();
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
```

```
primaryColor: Colors.indigo[300],
canvasColor: Colors.indigo[900],
visualDensity: VisualDensity.adaptivePlatformDensity,
tooltipTheme: TooltipThemeData(
  decoration: ShapeDecoration(
    color: Color(0xFF232F34),
    shape: StadiumBorder(),
  ),
),
)),
home: FutureBuilder<bool>(
  future: isRegister(),
  builder: (BuildContext context, AsyncSnapshot<bool> snapshot) {
    if (snapshot.data == true) {
      return BottomNavigationPage();
    } else {
      return SignInPage();
    }
  },
),
);
}
Future _showNotification(String title, String body) async {
  var androidDetails = new AndroidNotificationDetails("channelId",
"channelName", "channelDescription");
  var iosDetails = new IOSNotificationDetails();
  var generalNotificationsDetails = new NotificationDetails(android: androidDetails,
iOS: iosDetails);
  await localNotificationsPlugin.show(0, title, body, generalNotificationsDetails);
}
void iOS_Permission() {
  _firebaseMessaging.requestNotificationPermissions(
    IosNotificationSettings(sound: true, badge: true, alert: true)
  );
  _firebaseMessaging.onIosSettingsRegistered
    .listen((IosNotificationSettings settings)
  {
    // print("Settings registered: $settings");
  });
}
_getToken() async {
  SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
  _firebaseMessaging.getToken().then((String token) async {
    assert(token != null);
  });
}
```



```
print(token);

    sharedPreferences.setString(AppConstants.deviceToken, token);
    });
}
_configureFirebaseListeners(BuildContext context) {
    _firebaseMessaging.configure(
        onMessage: (Map<String, dynamic> message) async {
            print(message);
            Map<String, dynamic> convertedMessage = _convertMessage(message);
            if(convertedMessage != null) {
                print("onMessage: $message");
                _showNotification(convertedMessage["title"], convertedMessage["body"]);
            }
        },
        onLaunch: (Map<String, dynamic> message) async {
            print(message);
            Map<String, dynamic> convertedMessage = _convertMessage(message);
            if(convertedMessage != null) {
                print("onLaunch: $message");
            }
        },
        onResume: (Map<String, dynamic> message) async {
            print(message);
            Map<String, dynamic> convertedMessage = _convertMessage(message);
            if(convertedMessage != null) {
                print("onResume: $message");
            }
        },
    );
}

Map<String, dynamic> _convertMessage(Map<String, dynamic> message) {
    try {
        if (Platform.isIOS) {
            return {
                'title': 'Agregator',
                'body': message['aps']['alert'],
            };
        } else {
            return {
                'title': 'Agregator',
```

```
        'body': message['notification']['body'],
      };
    }

    } catch (e) {
      return null;
    }
  }
}
```

2. *sign\_in\_page.dart* – Авторизациялау беті.

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:flutter/material.dart';
import 'package:pharmacy_aggregator/components/appBar.dart';
import 'package:pharmacy_aggregator/core/constants.dart';
import
'package:pharmacy_aggregator/pages/authorization/password_recovery_page.dart';
import 'package:pharmacy_aggregator/pages/authorization/sign_up_page.dart';
import 'package:pharmacy_aggregator/utils/utils.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../bottom_navigation_page.dart';
class SignInPage extends StatefulWidget {
  @override
  _SignInPageState createState() => _SignInPageState();
}
class _SignInPageState extends State<SignInPage> {
  var menu = ['Авторизация', 'Редактировать профиль', 'Служба
поддержки', 'Настройки', 'Выход'];
  TextEditingController loginController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  final GlobalKey<ScaffoldState> _scaffoldKey = new GlobalKey<ScaffoldState>();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      key: _scaffoldKey,
      appBar: buildAppBar(""),
      backgroundColor: Colors.white,
```

```
body: Container(child:
  Column(
    crossAxisAlignment: CrossAxisAlignment.center,
    children: [

      Center(child: Image.asset('assets/images/Title.png', width:
MediaQuery.of(context).size.width*0.9,)),
      SizedBox(height: 30),
      Container(
        decoration: BoxDecoration(color: Colors.indigo[900], borderRadius:
BorderRadius.circular(20)),
        width: MediaQuery.of(context).size.width*0.85,
        child: Padding(
          padding: const EdgeInsets.all(30.0),
          child: Column(
            children: [
              Text('Авторизация', style: TextStyle(color: Colors.white, fontSize: 22,
fontFamily: 'Montserrat Regular'),),
              SizedBox(height: 20),
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Padding(
                    padding: const EdgeInsets.symmetric(horizontal: 10.0),
                    child: Text('Логин', style: TextStyle(color: Colors.white, fontSize: 16,
fontFamily: 'Montserrat Regular')),
                  ),
                  Padding(
                    padding: const EdgeInsets.symmetric(horizontal: 10.0, vertical: 10),
                    child: TextFormField(
                      keyboardType: TextInputType.text,
                      controller: loginController,
                      decoration: InputDecoration(
                        filled: true,
                        fillColor: Colors.white
                      ),
                    ),
                  ),
                ],
              ),
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 10.0),
                child: Text('Пароль', style: TextStyle(color: Colors.white, fontSize: 16,
fontFamily: 'Montserrat Regular')),
              ),
            ],
          ),
        ),
      ),
    ],
  ),
),
```

```

Padding(
  padding: const EdgeInsets.symmetric(horizontal: 10.0, vertical: 10),
  child: TextFormField(
    obscureText: true,
    keyboardType: TextInputType.text,
    controller: passwordController,
    decoration: InputDecoration(
      filled: true,
      fillColor: Colors.white
    ),
  ),
),
),
),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Center(
    child: GestureDetector(
      child: Text('Забыли пароль?', style: TextStyle(color: Colors.white70,
fontSize: 14, fontFamily: 'Montserrat Regular', decoration:
TextDecoration.underline,)),
      onTap: () {
        Navigator.push(context, MaterialPageRoute(builder: (BuildContext
context) => PasswordRecoveryPage()));
      },
    ),
  ),
),
),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Center(
    child: ButtonTheme(
      minWidth: 200,
      height: 50,
      child:
        RaisedButton(
          onPressed: () {
            var isConnected = checkInternetConnection();
            isConnected.then((value) => {
              if (value) {
                createAlertDialog(context),
                reg(loginController.text,
                  passwordController.text.trim())
              }
            })
          }
        )
      ),
  ),
),
),

```

```
else{
  showAlert(
    "Внимание",
    "У вас нет соединения с интернетом!",
    context)
}

});
FocusScopeNode currentFocus =
  FocusScope.of(context);
if (!currentFocus.hasPrimaryFocus) {
  currentFocus.unfocus();
}
},
shape: RoundedRectangleBorder(
  borderRadius: BorderRadius.circular(30)
),
padding: const EdgeInsets.symmetric(vertical: 10.0),
textColor: Colors.white,
color: Colors.lightBlueAccent,
elevation: 3.0,
child: Text(
  "Войти",
  style: TextStyle(
    fontSize: 16,
  ),
),
),
),
),
),
),
),
),
),
Padding(
  padding: const EdgeInsets.all(12.0),
  child: Center(
    child: GestureDetector(
      child: Text('Регистрация', style: TextStyle(color: Colors.white,
fontSize: 14, fontFamily: 'Montserrat Regular', decoration:
TextDecoration.underline), ),
    onTap: (){
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => SignUpPage()) //login,
```

```

password ,
        );
        },
        ),
        ),
        ],),
    ],
    )
    ),
    ),
    ],)),
);
}
void reg(String login, String password) async {
    print(login);
    print(password);
    String jsonString = await logIn(
        login,
        password,
    );
    print(jsonString);
    if (jsonString != null){
        Map<String, dynamic> status = jsonDecode(jsonString);
        print(status);

        if (status['uid'] != null) {
            SharedPreferences prefs = await SharedPreferences.getInstance();
            prefs.setBool('isReg', true);
            prefs.setString('Token', status['key']);
            prefs.setInt('uid', status['uid']);
            Navigator.pop(context);
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (BuildContext context) => BottomNavigationBar() //login,
password ,
                );
            } else {
                Navigator.pop(context);
                _scaffoldKey.currentState.showSnackBar(SnackBar(
                    content: Text(

```



```
        'Проверьте соединения с интернетом или попробуйте позже!',
    ));
  }
}
else{
  Navigator.pop(context);
  _scaffoldKey.currentState.showSnackBar(SnackBar(
    content: Text(
      'Неверный логин или пароль!',
    )),
  )
}
Future<String> logIn(String login, String password) async {
  final response = await http.post(AppConstants.baseUrl + "users/login/",
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
    },
    body: jsonEncode(<String, String>{
      'username': login,
      'password': password,
    }));
  print(response.statusCode);
  if (response.statusCode == 200) {
    print(response.body);
    return response.body;
  } else {
    return null;
  }
}
}
```

## 2. *sign\_up\_page.dart* – Тіркелу беті.

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:flutter/material.dart';
import 'package:pharmacy_aggregator/components/appBar.dart';
import 'package:pharmacy_aggregator/core/constants.dart';
import
'package:pharmacy_aggregator/pages/authorization/password_recovery_page.dart';
import 'package:pharmacy_aggregator/utils/utils.dart';
```

```
import 'package:shared_preferences/shared_preferences.dart';
import '../bottom_navigation_page.dart';
class SignUpPage extends StatefulWidget {
  @override
  _SignUpPageState createState() => _SignUpPageState();
}
class _SignUpPageState extends State<SignUpPage> {
  TextEditingController loginController = TextEditingController();

  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();

  final GlobalKey<ScaffoldState> _scaffoldKey = new GlobalKey<ScaffoldState>();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      key: _scaffoldKey,
      appBar: buildAppBar(""),
      backgroundColor: Colors.white,
      body: Container(child:
        Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Center(child: Image.asset('assets/images/Title.png', width:
MediaQuery.of(context).size.width*0.9)),
            SizedBox(height: 10),
            Container(
              decoration: BoxDecoration(color: Colors.indigo[900], borderRadius:
BorderRadius.circular(20)),
              width: MediaQuery.of(context).size.width*0.85,
              child: Padding(
                padding: const EdgeInsets.all(10.0),
                child: Column(
                  children: [
                    SizedBox(height: 20),
                    Text('Регистрация', style: TextStyle(color: Colors.white, fontSize: 22,
fontFamily: 'Montserrat Regular'),),
                    SizedBox(height: 10),
                    Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Padding(
```

```
padding: const EdgeInsets.symmetric(horizontal: 10.0),
child: Text('Логин', style: TextStyle(color: Colors.white, fontSize: 16,
fontFamily: 'Montserrat Regular')),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 10.0, vertical: 10),
child: TextFormField(
keyboardType: TextInputType.text,
controller: loginController,

decoration: InputDecoration(
filled: true,
fillColor: Colors.white
),
),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 10.0),
child: Text('Email', style: TextStyle(color: Colors.white, fontSize: 16,
fontFamily: 'Montserrat Regular')),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 10.0, vertical: 10),
child: TextFormField(
keyboardType: TextInputType.text,
controller: emailController,
decoration: InputDecoration(
filled: true,
fillColor: Colors.white
),
),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 10.0),
child: Text('Пароль', style: TextStyle(color: Colors.white, fontSize: 16,
fontFamily: 'Montserrat Regular')),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 10.0, vertical: 10),
child: TextFormField(
obscureText: true,
keyboardType: TextInputType.text,
controller: passwordController,
```

```
decoration: InputDecoration(
  filled: true,
  fillColor: Colors.white
),
),
),
Padding(
  padding: const EdgeInsets.all(40.0),
  child: Center(
    child: ButtonTheme(

      minWidth: 200,
      height: 50,
      child:

        RaisedButton(
          onPressed: () {
            var isConnected = checkInternetConnection();
            isConnected.then((value) => {
              if (value){
                createAlertDialog(context),
                reg(loginController.text, emailController.text,
                  passwordController.text.trim())
              }
              else{
                showAlert(
                  "Внимание",
                  "У вас нет соединения с интернетом!",
                  context)
              }
            });
            FocusScopeNode currentFocus =
              FocusScope.of(context);
            if (!currentFocus.hasPrimaryFocus) {
              currentFocus.unfocus();
            }
          },
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(30)
          ),
          padding: const EdgeInsets.symmetric(vertical: 10.0),
          textColor: Colors.white,
          color: Colors.lightBlueAccent,
```

```
elevation: 3.0,
child: Text(
  "Зарегистрироваться",
  style: TextStyle(
    fontSize: 16,
  ),
),
),
),
),
),
),
),
],),
],
),
),
],)),
);
}
void reg(String login, String email, String password) async {
  String jsonString = await registration(
    login,
    email,
    password,
  );
  Map<String, dynamic> status = jsonDecode(jsonString);
  print(status);
  if (status['status'] == "ok") {
    Navigator.pop(context);
    _scaffoldKey.currentState.showSnackBar(SnackBar(
      content: Text(
        'Регистрация прошла успешно, войдите в свою учетную запись!',
      )),);
  } else {
    Navigator.pop(context);
    _scaffoldKey.currentState.showSnackBar(SnackBar(
      content: Text(
        'Проверьте соединения с интернетом или попробуйте позже!',
      )),);
    // print("status no ok");
  }
}
```

```
Future<String> registration(String login,String email, String password) async {
  final response = await http.post(AppConstants.baseUrl + "users/register/",
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
    },
    body: jsonEncode(<String, String>{
      'username': login,
      'email': email,
      'password': password,
    }));
  if (response.statusCode == 200) {
    return response.body;

  } else {
    throw Exception("Falied to registration");
  }
}
```

*3. Home\_page.dart – Бастапқы негізгі бет.*

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:pharmacy_aggregator/components/appBar.dart';
import 'package:pharmacy_aggregator/core/constants.dart';
import 'package:pharmacy_aggregator/models/medication.dart';
import 'package:pharmacy_aggregator/pages/home/search_page.dart';
import 'package:pharmacy_aggregator/pages/medication/medication_data.dart';
import 'package:pharmacy_aggregator/pages/medication/medication_page.dart';
import 'package:pharmacy_aggregator/utills/utills.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:convert';
import 'package:http/http.dart' as http;
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  TextEditingController _searchController = new TextEditingController();
  FocusNode focusNode = new FocusNode();
  List<Medication> medicationList = [];
  @override
  void initState() {
```

```
super.initState();
getProducts();
sendDeviceToken();
focusNode.unfocus();
}
@override
void dispose() {
  // TODO: implement dispose
  super.dispose();
  focusNode.dispose();
}
@override
Widget build(BuildContext context) {

  var size = MediaQuery.of(context).size;
  final double itemHeight = (size.height - kToolbarHeight - 24) / 2;
  final double itemWidth = size.width / 2;
  return Scaffold(

    appBar: buildAppBar("Главная"),
    backgroundColor: Colors.white,
    body: SingleChildScrollView(
      child: Container(
        padding: EdgeInsets.symmetric(horizontal: 10, vertical: 2),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            getText("Введите название лекарства", Icons.search),
            Padding(
              padding: const EdgeInsets.only(top: 14),
              child: Text("Категории лекарств",
                style: TextStyle(
                  fontSize: 16,
                  fontFamily: AppFonts.montessoratSemiBold,
                  color: Colors.black)),
            ),
            Padding(
              padding: const EdgeInsets.only(top: 12, left: 5, right: 5),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  CircleButton(
                    onTap: () =>
```



```
Navigator.push(  
  context,  
  MaterialPageRoute(  
    builder: (context) => MedicationPage(1))),  
  iconData: 'assets/images/1.png',  
  titleText: "Лекарственные препараты"),  
CircleButton(  
  onTap: () =>  
    Navigator.push(  
      context,  
      MaterialPageRoute(  
        builder: (context) => MedicationPage(2))),  
  iconData: 'assets/images/2.png',  
  titleText: "Медицинские изделия и приборы"),  
  
CircleButton(  
  onTap: () =>  
    Navigator.push(  
      context,  
      MaterialPageRoute(  
  
        builder: (context) => MedicationPage(3))),  
  iconData: 'assets/images/3.png',  
  titleText: "Травы, сборы, бальзамы"),  
  ],  
),  
),  
Padding(  
  padding: const EdgeInsets.only(top: 12, left: 5, right: 5),  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: [  
      CircleButton(  
        onTap: () =>  
          Navigator.push(  
            context,  
            MaterialPageRoute(  
              builder: (context) => MedicationPage(4))),  
        iconData: 'assets/images/4.png',  
        titleText: "Витамины и БАДы"),  
      CircleButton(  
        onTap: () =>  
          Navigator.push(  

```

```
        context,
        MaterialPageRoute(
            builder: (context) => MedicationPage(5))),
        iconData: 'assets/images/5.png',
        titleText: "Косметика и средства гигиены"),
    CircleButton(
        onTap: () =>
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => MedicationPage(6))),
        iconData: 'assets/images/6.png',
        titleText: "Мама и малыш"),
    ],
),
),
Padding(
    padding: const EdgeInsets.only(top: 14),
    child: Text("Рекомендуем",
        style: TextStyle(
            fontSize: 16,

            fontFamily: AppFonts.montesseratSemiBold,
            color: Colors.black)),
),
Padding(
    padding: const EdgeInsets.only(top: 12, left: 5, right: 5),
    child: SizedBox(
        height: (medicationList.length~/3+1)*itemWidth,
        child: GridView.count(
            crossAxisSpacing: 5,
            mainAxisSpacing: 5,
            crossAxisCount: 3,
            physics: new NeverScrollableScrollPhysics(),
            childAspectRatio: (itemWidth / itemHeight),
            children: List.generate(medicationList.length, (index) {
                return CircleImage(
                    onTap: () => {
                        Navigator.push(context,
                            MaterialPageRoute(builder: (context) =>
MedicationData(medicationList[index])))
                    },

```

```
        imageURL: medicationList[index].img,
        titleText: medicationList[index].name
    );
    }),
    ),
    ),
    Padding(
        padding: const EdgeInsets.only(
            top: 12, left: 5, right: 5, bottom: 8),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
        ),
    )
],
),
),
);
}
Widget getText(String text, IconData icon) {
    return Padding(
        padding: const EdgeInsets.fromLTRB(6, 5, 6, 5),

        child: GestureDetector(
            onTap: () {
                Navigator.push(
                    context, MaterialPageRoute(builder: (context) => SearchPage()));
            },
            child: Container(
                width: MediaQuery
                    .of(context)
                    .size
                    .width,
                decoration: BoxDecoration(
                    border: Border.all(width: 1, color: Colors.black45),
                    borderRadius: BorderRadius.all(Radius.circular(10.0)),
                    color: Colors.white),
                child: Padding(
                    padding: const EdgeInsets.fromLTRB(10, 13, 10, 13),
                    child: Row(
                        mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```
children: [
  Text(text, style: TextStyle(color: Colors.black,
    fontFamily: AppFonts.montserratRegular)),
  Padding(
    padding: const EdgeInsets.only(right: 8.0),
    child: Icon(icon, color: Colors.black),
  ),
],
),
)),
),
);
}
sendDeviceToken() async {
  SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
  var isSended =
  sharedPreferences.getBool(AppConstants.isSendedDeviceToken);

  if (isSended == null || isSended == false) {
    var token = sharedPreferences.getString("Token");
    var deviceToken = sharedPreferences.getString(AppConstants.deviceToken);
    // print(deviceToken);
    if (deviceToken != null){
      var url = "${AppConstants.baseUrl}users/push/register/";
      var headers = {
        "Accept": "application/json",
        "Authorization": "Token $token"
      };
      final response = await http.post(url, headers: headers, body: {
        "reg_id": deviceToken,
        // "cmt": "fcm",
      });
      if (response.statusCode == 200) {
        Map<String, dynamic> status = jsonDecode(response.body);
        if (status['status'] == "ok") {
          print("send device token");
          sharedPreferences.setBool(AppConstants.isSendedDeviceToken, true);
          // print("sended device token");
        }
      }
    }
  }
}
```

```

getProducts() async {
  SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
  var token = sharedPreferences.getString("Token");
  await http.get(
    "${AppConstants.baseUrl}product/recomendation/",
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
      "Accept": "application/json",
      "Authorization": "Token $token"
    },
  ).then((response) {
    List<Medication> list = List<Medication>();
    var responseBody = jsonDecode(utf8.decode(response.body.codeUnits));
    // print(responseBody);
    for (Object i in responseBody){
      Map<String,dynamic> j = i;
      if (j['available'].length != 0){
        list.add(Medication(j['id'], j['name'], j['manufacturer']['name'], j['photo'] ,
j['description'], 'от ' + j['available'][0]['price'].toString() + 'тг.', true, j['composition'],
j['available']));
      }
      else{
        list.add(Medication(j['id'], j['name'], j['manufacturer']['name'], j['photo'] ,
j['description'], 'от 0 тг.', true, j['composition'], j['available']));
      }
    }
    setState() {
      medicationList = list;
    });
  }).catchError((error) => print(error));
}
}

class CircleImage extends StatelessWidget {
  final GestureTapCallback onTap;
  final String imgURL;
  final String titleText;
  const CircleImage({Key key, this.onTap, this.imgURL, this.titleText})
    : super(key: key);
  @override
  Widget build(BuildContext context) {
    double size = MediaQuery
      .of(context)

```

```
.size
.width / 3.7;
return Container(
  width: size,
  child: Column(
    children: [
      InkResponse(
        onTap: onTap,
        child: new Container(
          width: size,
          height: size,
          decoration: new BoxDecoration(
            shape: BoxShape.circle,
          ),
          child: CachedNetworkImage(
            imageUrl: imgUrl,
            imageBuilder: (context, imageProvider) => Container(
              // width: MediaQuery.of(context).size.width / 3,
              // height: MediaQuery.of(context).size.width / 3,

              decoration: BoxDecoration(
                image: DecorationImage(
                  image: imageProvider,
                  fit: BoxFit.cover,
                ),
              ),
            ),
            placeholder: (context, url) => CircularProgressIndicator(),
            errorWidget: (context, url, error) => Icon(Icons.error, size: size),
          ),
        ),
      ),
    ),
  ),
  Padding(
    padding: const EdgeInsets.only(top: 8),
    child: Text(titleText,
      textAlign: TextAlign.center,
      maxLines: 2,
      softWrap: true,
      overflow: TextOverflow.ellipsis,
      style: TextStyle(
        fontSize: 12,
        color: Colors.black,
        fontFamily: AppFonts.montserratMedium,
```

```
        fontWeight: FontWeight.bold)),
    ),
  ],
),
);
}
}
```

```
class CircleButton extends StatelessWidget {
  final GestureTapCallback onTap;
  final String iconData;
  final String titleText;

  const CircleButton({Key key, this.onTap, this.iconData, this.titleText})
    : super(key: key);

  @override
  Widget build(BuildContext context) {
    double size = MediaQuery
      .of(context)

      .size
      .width / 3.7;
    return Container(
      width: size,
      child: Column(
        children: [
          InkResponse(
            onTap: onTap,
            child: new Container(
              width: size,
              height: size,
              decoration: new BoxDecoration(
                color: Colors.indigo[900],
                shape: BoxShape.circle,
              ),
            ),
            child: new Image.asset(
              iconData,
            ),
          ),
        ],
      ),
      Padding(
        padding: const EdgeInsets.only(top: 8),

```

```
child: Text(titleText,  
  textAlign: TextAlign.center,  
  maxLines: 2,  
  softWrap: true,  
  overflow: TextOverflow.ellipsis,  
  style: TextStyle(  
    fontSize: 12,  
    color: Colors.black,  
    fontFamily: AppFonts.montserratMedium,  
    fontWeight: FontWeight.bold)),  
  ),  
],  
),  
);  
}  
}
```



ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ  
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

5B070400 - Есептеуіш техника және бағдарламамен қамтамасыз ету

Даулетканова Молдир Муратканқызы

**ҒЫЛЫМИ ЖЕТЕКШІНІҢ  
СЫН-ШІКІРІ**

**Тақырыбы:** «Дәріханалар бойынша мобильдік қолданба-агрегаторды әзірлеу».

Студент Даулетканова М. М алдына дәріханалар бойынша мобильдік қолданба-агрегаторын жасау міндеті қойылды. Бұл дипломдық жоба дәріханалардың қызметін мобильді қолданба - агрегаторын қолдану арқылы жеңілдету мақсатында құрылған.

Дипломдық жобаның бірінші бөлімінде мобильді қосымша жасаудағы өзектіліктер мен мақсаттына, ерекшеліктеріне талдау жасалынды. Сондай-ақ, ұқсас мобильді қосымшаларға шолу жүргізілді. Екінші бөлімде мобильді қосымшаның бағдарламалық тілі таңдалынып, программалық қамтамаларға қысқаша шолу жасалынды. Үшінші бөлім - негізгі бөлімі болып табылады. Бұл бөлімде қосымшаны жобалауға қатысты UML диаграммалар көрсетіліп, қолданушылар интерфейсіне қысқаша сипаттама жасалған. Даулетканова М. М. мобильді қосымшаны жобалау кезінде пайдаланушы интерфейсін мүмкіндігінше қарапайым және интуитивті түсінікті етіп жасап шықты.

Дипломдық жобаны студент қойылған тапсырма бойынша барлық талаптарға сай, тақырып төңірегінде жазған.

Қортындылай келе, студент өз мүмкіндіктерін кеңінен пайдаланып, дипломдық жобаны жақсы жасап шыққан.

Ғылыми жетекшісі ретінде дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып, Даулетканова Молдир Муратканқызы дипломдық жобаны қорғауға ұсынамын және 5B070400 – «Есептеуіш техника және программамен қамтамасыз ету» мамандығы бойынша «Бакалавр» академиялық деңгейін тағайындауға лайық деп есептеймін.

**ҒЫЛЫМИ ЖЕТЕКШІ**  **Баймбетов Д.А.**

**Техника ғылымдарының магистрі,**

**«Программалық инженерия»**

**Кафедрасының лекторы** « 21 » 05 2021 жыл



## Метаданные

Название

**Дипломная работа (Даулетканова Молдир).docx**

Автор

**Даулетканова Молдир**

Научный руководитель






**Даулет Байымбетов**

Подразделение

**ИКИИТ**

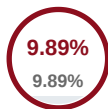
## Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		1
Интервалы		0
Микропробелы		0
Белые знаки		0
Парафразы (SmartMarks)		50

## Объем найденных подобиий

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.

**25**

Длина фразы для коэффициента подобия 2

**6858**

Количество слов

**72732**

Количество символов

## Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

### 10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	<a href="https://syrbayi.kz/janalyqtar/onir/daeri-daermek-qory-zhetkiilkti-2-62749/">https://syrbayi.kz/janalyqtar/onir/daeri-daermek-qory-zhetkiilkti-2-62749/</a>	30	0.44 %
2	<a href="https://teratail.com/questions/174577">https://teratail.com/questions/174577</a>	27	0.39 %
3	<b>ЖШС «J.I.T. GROUP» АҚПАРАТЫН ҚЫЗМЕТІН ЕСЕПКЕ АЛУДЫ АВТОМАТТАНДЫРУ (КОМПЬЮТЕРЛІК ЖӘНЕ МОБИЛЬДІК ҚОСЫМШАЛАРЫН ҚҰРАСТЫРУ)</b> Жиренбаева Рая 6/16/2020 Kazakh National Agrarian University (КазНАУ)	25	0.36 %
4	<b>Қоймадағы радиоэлементтерді есепке алатын ақпараттық жүйе</b> Қызыр Гүлнұр Даулетқызы 5/10/2018 Satbayev University (ИКИИТ)	18	0.26 %

5	<a href="https://zhuanlan.zhihu.com/p/41027831">https://zhuanlan.zhihu.com/p/41027831</a>	17	0.25 %
6	<a href="https://www.warmodroid.xyz/tutorial/flutter/learn-how-to-design-profile-details-ui-in-the-flutter/">https://www.warmodroid.xyz/tutorial/flutter/learn-how-to-design-profile-details-ui-in-the-flutter/</a>	17	0.25 %
7	<b>Мыстың цементация процесін зерттеу</b> Сулеев Ербота Әбдіхалықұлы 5/15/2018 Satbayev University (Г_М_И)	16	0.23 %
8	<a href="https://here4you.tistory.com/225">https://here4you.tistory.com/225</a>	15	0.22 %
9	<a href="https://zhuanlan.zhihu.com/p/41027831">https://zhuanlan.zhihu.com/p/41027831</a>	14	0.20 %
10	<a href="https://stud.kz/referat/show/62961">https://stud.kz/referat/show/62961</a>	13	0.19 %

#### из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

#### из домашней базы данных (0.50 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	<b>Қоймадағы радиоэлементтерді есепке алатын ақпараттық жүйе</b> Қызыр Гүлнұр Даулетқызы 5/10/2018 Satbayev University (ИКИИТ)	18 (1) 0.26 %
2	<b>Мыстың цементация процесін зерттеу</b> Сулеев Ербота Әбдіхалықұлы 5/15/2018 Satbayev University (Г_М_И)	16 (1) 0.23 %

#### из программы обмена базами данных (0.69 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	<b>ЖШС «J.I.T. GROUP» АҚПАРАТЫН ҚЫЗМЕТІН ЕСЕПKE АЛУДЫ АВТОМАТТАНДЫРУ (КОМПЬЮТЕРЛІК ЖӘНЕ МОБИЛЬДІК ҚОСЫМШАЛАРЫН ҚҰРАСТЫРУ)</b> Жиренбаева Рая 6/16/2020 Kazakh National Agrarian University (КазНАУ)	25 (1) 0.36 %
2	<b>Банктің қаржылық жағдайын бағалаудың компьютерлік үлгісін құру</b> Амангелдиева Ақерке Ұланқызы 5/23/2021 D. Serikbayev East Kazakhstan State Technical University (Админ)	22 (3) 0.32 %

#### из интернета (8.71 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	<a href="https://here4you.tistory.com/225">https://here4you.tistory.com/225</a>	205 (28) 2.99 %
2	<a href="https://github.com/Widle-Studio/Grocery-App/blob/master/f_groceries/lib/item_screen.dart">https://github.com/Widle-Studio/Grocery-App/blob/master/f_groceries/lib/item_screen.dart</a>	55 (8) 0.80 %
3	<a href="https://teratail.com/questions/174577">https://teratail.com/questions/174577</a>	51 (5) 0.74 %
4	<a href="https://www.warmodroid.xyz/tutorial/flutter/learn-how-to-design-profile-details-ui-in-the-flutter/">https://www.warmodroid.xyz/tutorial/flutter/learn-how-to-design-profile-details-ui-in-the-flutter/</a>	41 (4) 0.60 %
5	<a href="https://zhuanlan.zhihu.com/p/41027831">https://zhuanlan.zhihu.com/p/41027831</a>	31 (2) 0.45 %
6	<a href="https://syroyoi.kz/janalyqtar/onir/daeri-daermek-qory-zhetkilikti-2-62749/">https://syroyoi.kz/janalyqtar/onir/daeri-daermek-qory-zhetkilikti-2-62749/</a>	30 (1) 0.44 %
7	<a href="https://blog.codemagic.io/building-responsive-applications-with-flutter/">https://blog.codemagic.io/building-responsive-applications-with-flutter/</a>	30 (4) 0.44 %

8	<a href="https://pub.dev/documentation/cached_network_image/latest/">https://pub.dev/documentation/cached_network_image/latest/</a>	21 (2)	0.31 %
9	<a href="https://www.filledstacks.com/post/building-a-responsive-website-using-flutter/">https://www.filledstacks.com/post/building-a-responsive-website-using-flutter/</a>	20 (3)	0.29 %
10	<a href="https://github.com/flutter/flutter/issues/64377">https://github.com/flutter/flutter/issues/64377</a>	20 (3)	0.29 %
11	<a href="https://cloudstack.ninja/rishabh/packagecached_network_image-src-cached_network_image_provider-dart-failed-assertion-line-18-pos-16-url-null-is-not-true/">https://cloudstack.ninja/rishabh/packagecached_network_image-src-cached_network_image_provider-dart-failed-assertion-line-18-pos-16-url-null-is-not-true/</a>	16 (2)	0.23 %
12	<a href="https://stackoverflow.com/questions/55320339/how-to-make-a-basic-http-auth-login-and-register-and-store-the-token">https://stackoverflow.com/questions/55320339/how-to-make-a-basic-http-auth-login-and-register-and-store-the-token</a>	14 (2)	0.20 %
13	<a href="https://stud.kz/referat/show/62961">https://stud.kz/referat/show/62961</a>	13 (1)	0.19 %
14	<a href="https://blog.codemagic.io/flutter-web-getting-started-with-responsive-design/">https://blog.codemagic.io/flutter-web-getting-started-with-responsive-design/</a>	12 (2)	0.17 %
15	<a href="https://blog.faradars.org/create-a-simple-login-form-in-flutter-using-bloc-pattern/">https://blog.faradars.org/create-a-simple-login-form-in-flutter-using-bloc-pattern/</a>	11 (1)	0.16 %
16	<a href="https://www.filledstacks.com/snippet/build-a-flutter-location-service/">https://www.filledstacks.com/snippet/build-a-flutter-location-service/</a>	8 (1)	0.12 %
17	<a href="https://stackoverflow.com/questions/52164369/show-alert-dialog-on-app-main-screen-load-automatically-in-flutter">https://stackoverflow.com/questions/52164369/show-alert-dialog-on-app-main-screen-load-automatically-in-flutter</a>	7 (1)	0.10 %
18	<a href="https://qstack.jp/programming/48405123/how-to-set-custom-height-for-widget-in-gridview-in-flutter">https://qstack.jp/programming/48405123/how-to-set-custom-height-for-widget-in-gridview-in-flutter</a>	6 (1)	0.09 %
19	<a href="https://stackoverflow.com/questions/50602425/how-to-set-grid-view-column-height">https://stackoverflow.com/questions/50602425/how-to-set-grid-view-column-height</a>	6 (1)	0.09 %

### Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Даулетканова Молдир Муратканқызы

**Название:** Дәріханалар бойынша мобильдік қолданба-агрегаторды әзірлеу

**Координатор:** Сейтбекова Е.С.

**Коэффициент подобия 1:** 9.89%

**Коэффициент подобия 2:** 1.20%

**Замена букв:** 1

**Интервалы:** 0

**Микропробелы:** 3

**Белые знаки:** 0

### После анализа Отчета подобия констатирую следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....  
.....  
.....  
.....

Дата 26.05.2021

Подпись Научного руководителя



<i>Форматы</i>	<i>Зонасы</i>	<i>Поз-сы</i>	<i>Белгілеулер</i>	<i>Атаулары</i>	<i>Саны</i>	<i>Қоры- тынды</i>
<i>Өлш.</i>	<i>Бет</i>	<i>Құжат №</i>	<i>Қолы</i>	<i>Күні</i>	<i>Дипломдық жоба</i>	
						<i>Бет</i> 56