

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Азберген Төребек Қуатұлы

Бос күндеріңізді өткізуге көмектесетін мобильді қосымшаны әзірлеу

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд, профессор

А.Н.Молдагулова

" " 2022 ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: «Бос күндеріңізді өткізуге көмектесетін мобильді қосымшаны
әзірлеу»

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

Орындаған

Азберген Т. Қ.

Рецензент

Страший преподаватель

Ғылыми жетекші

Магистр, лектор

Уалиева И. М.

" 26 " 05 2022 ж.

Бекарыстанқызы А..

" 25 " 05 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН

ПИ кафедрасының меңгерушісі

физ.-мат.ғыл.канд, профессор

_____ А.Н. Молдагулова

« _____ » _____ 2022 г.

Дипломдық жобаны орындауға

ТАПСЫРМА

Білім алушыға *Азберген Төребек Қуатұлы*

Тақырыбы: «Бос күндеріңізді өткізуге көмектесетін мобильді қосымшаны әзірлеу»

Академиялық мәселелер жөніндегі проректор бұйрығының
№ *489-III/24* *12* 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі: " *26* " *05* 2022 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 қаматамаларды жасау жүргізілген.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

- а) негізгі бөлім;
- б) жүйені модельдеу;
- в) техникалық бөлім;
- г) жобалау бөлімі;
- д) деректермен жұмыс бөлімі;
- е) mail шаблон бөлімі;

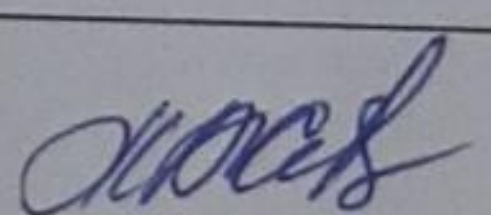
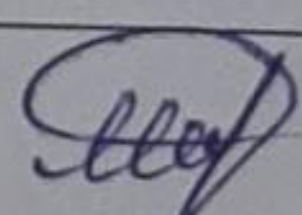
Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен): презентацияның 14 слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер: 20 пайдаланылған әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Пән саласын талдау, техникалық тапсырма әзірлеу	3.03.2022	орындалды
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	11.03.2022	орындалды
3. Дипломдық жобаның веб қосымшасын дайындау және іске асыру	20.03.2022	орындалды
4. Жобаның бөліміне сызбаларды дайындау.	01.04.2022	орындалды
5. Жүйе функционалдығын дамыту	16.04.2022	орындалды
6. Тестілеу және безендіру	19.04.2022	орындалды
7. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	28.04.2022	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

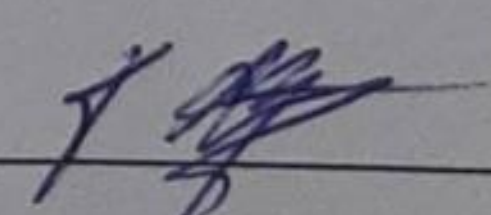
Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Жекамбаева М.Н. Ph.D. доктор, қауымдастырылған профессор	23.05.22	
Бағдарламалық бөлім	Марғұлан Қ., тех. ғыл. магистірі, лектор	23.05.22	

Ғылыми жетекші



Бекарыстанқызы А..

Тапсырманы орындауға қабылдаған студент



Азберген Т. Қ.

Күні

« 14 » 11 2021ж.

АҢДАТПА

Дипломдық жұмыстағы мақсаты бұл үйілген істерден ең маңыздыларын бөліп алып және соларды қадағалау. Күнделікті әрбір секундты мұқият жоспарлау арқылы уақытты ақылмен басқара отырып, уақытыңызды тиімді өткізу үшін және соңында мақсаттарыңызға жете алуына көмектеседі. Уақытты дұрыс басқару әдетте ұйымдаспауға, мерзімдерді өткізіп алуға және бос уақыттың аздығына әкеледі.

Уақытты дұрыс басқарудың негізгі себептері - ұқыптылық пен басымдықтың болмауы және жұмысты белгісіз мерзімге кейінге қалдыруы.

АННОТАЦИЯ

Целью моей дипломной работы является выявление и мониторинг наиболее важных дел. Тщательное планирование каждой секунды каждого дня поможет вам разумно распоряжаться своим временем, тратить его эффективно и в конечном итоге достигать поставленных целей. Правильное управление временем обычно приводит к неорганизованности, пустой трате времени и нехватке свободного времени.

Основными причинами плохого тайм-менеджмента являются отсутствие пунктуальности и приоритета, а задержки работают на неопределенный срок.

ANNOTATION

The purpose of my thesis is to identify and monitor the most important cases. Carefully planning every second of every day will help you manage your time wisely, spend it efficiently, and ultimately achieve your goals. Proper time management usually results in disorganization, waste of time and lack of free time.

The main reasons for poor time management are lack of punctuality and priority, and delays run indefinitely.

МАЗМҰНЫ

	Кіріспе	9
1	Негізгі бөлім	10
1.1	Жобаның мақсаты	10
1.2	Терминдер мен қысқартулар	10
1.3	Жобаның өзектілігі	11
1.4	Жүйені модельдеу	11
1.4.1	ER диаграммасы	11
1.4.2	USE Case диаграммасы	12
1.5	UML диаграммасы	13
1.5.1	Класс диаграммасы	13
1.6	Технологиялар бөлімі	14
1.6.1	Android studio	14
1.6.2	Бағдарламау тілі	15
1.6.3	Flutter фреймворкі	16
1.6.4	Figma	18
1.6.5	Firebase	18
2	Жобаның құрылымы	21
2.1	Аутентфикация бөлімі	21
2.2	Қолданушыдың басты бет көрінісі	22
2.3	Ескертпе қосу беті	23
2.4	Секундомер беті	24
2.5	Күнтізбе беті	25
3	Firebase деректермен жұмыс бөлімі	27
3.1	Firebase аутентфикация бөлімі	27
3.2	Firebase пайдаланушының деректер қорымен жұмыс бөлімі	27
3.2.1	Ескертпе деректері	27
3.2.2	Күнтізбе ескертпелері	28
3.2.3	Ескертпе ережесі	29
3.2.4	Ескертпенің қолдануы	29
4	Пошта үлгілері	30
4.1	Электрондық пошта мекенжайын тексеру	30
4.2	Құпия сөзді қалпына келтіру	30
4.3	Электрондық пошта мекенжайын өзгерту	31
4.4	SMS растау	31
	ҚОРЫТЫНДЫ	33
	ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	34
	Қосымша А БАҒДАРЛАМА МӘТІНІ	35

КІРІСПЕ

Бұл дипломдық жұмыста уақытты пайдалану және ұйымдастыру туралы шешім қабылдау үшін мобильді бағдарламасы жобаланды. Уақытты тиімді басқару бізге күнді тиімді пайдалануға және тапсырмаларды тезірек орындауға мүмкіндік береді. Қазіргі кезде осындай мобильді қосымшалар қарқынды дамуда. Уақытты тиімді басқару арқылы біз жұмыс пен өмірдің дұрыс басқаруға қол жеткізуге болады.

Қазіргі уақытта уақытты тиімді пайдалуыны үшін бірнеше техникалар бар. Соның бірі анализ парето техникасы мен помодоро техникасы. Парето техникасында басты идеясы 20 пайызы ол іс әрекет және 80 пайызы бұл нәтиже, ал помодоро техникасында жұмысыңызды аралықтарға бөлу үшін таймерді пайдаланады. Көбісінде 25 минут бұл жұмыс уақытына береді, ал 5 минут ол демалуға береді. Осындай техникалар арқасында уақытты тиімді пайдалануға көмектеседі.

Бұл техникаларды қолдану үшін бұрында дәптер, қалам, ескертпелер, сағат және тағыда басқа құралдар қажет етеін. Мобильді қосымшаны қолдана отырып барлық құрал саймандарды алмастыра алады және ол әрдайым сенін қолында. Сонымен қатар, жазбаларды басқа біреумен бөлісуге, оңай алуға мүмкін бар.

1 Негізгі бөлім

1.1 Жобаның мақсаты

Мобильді қосымшаны қолдана отырып, барлық маңызды жаңалықтар мен еске салғыштарды жазбаларды қолдана отырып жазуға болады. Мобильді ұйымдастырушы тізім жасау үшін өте қолайлы - жасалатын істер тізімі, демалыс жоспарлары және т.б. Сол себептен бұл қосымшалар басқаруға мүмкіндік беретін жеке көмекші ретінде жарамды.

1.2 Терминдер мен қысқартулар

Дипломдық жобада қолданылған терминдер мен аббревиатуралар төмендегі кестеде келтірілген.

1.1-кесте – Терминдер анықтамасы

Терминдер немесе қысқартулар	
UML	Жалпы мақсаттағы модельдеу тілі
framework	Бұл бағдарламалық жасақтаманы құруға болатын құрылым
Flutter	бұл бір код базасынан әдемі, жергілікті құрастырылған көп платформалы қолданбаларды құруға арналған Google ұсынған ашық бастапқы негіз.
Dart	Google жасаған бағдарламалау тілі
Пакет	қатысты кластар мен интерфейстер жиынын ұйымдастыратын аттар кеңістігі болып табылады
кітапхана	Кейбір ортада жалпы пайдалану үшін қолжетімді бағдарламалар мен бумалардың жиынтығы
Android	бұл мобильді операциялық жүйе
Ios	Apple компаниясының мобильді операциялық жүйесі
UI	бұл адамның компьютермен әрекеттесу және құрылғыдағы байланыс нүктесі.
Деректер базасы	оңай қол жеткізу, басқару және жаңартулар үшін теңшелген ақпарат
Firebase	мобильді және веб-қосымшаларды әзірлеу платформасы болып табылады

1.1 кесте - жалғасы

Json	деректерді құрылымдау үшін ең аз оқылатын пішім
анимация	қозғалыссыз сызбалар, позалар немесе кескіндер сериясы арқылы жасалған қозғалыс иллюзиясы
Use case	жүйелік талдауда қолданылатын әдістеме
IDE	қосымшаларды құруға арналған бағдарлама
GitHub	веб-хостинг қызметі

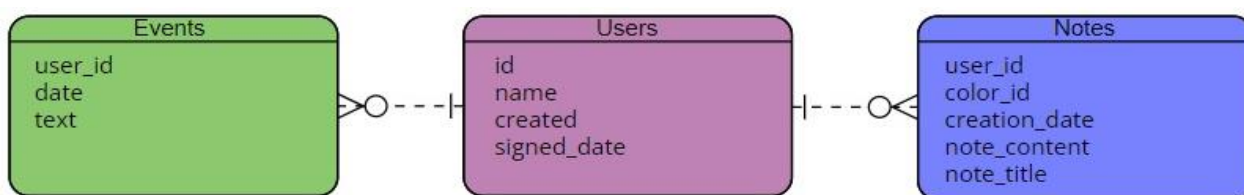
1.3 Жобаның өзектілігі

Мобильді қосымшаны пайдалана отырып, көп уақытты үнемдеуге, аз уақыт жұмсауға және өз қызметінде жақсы нәтижелерге қол жеткізуге көмектеседі. Уақытты басқару белгілі бір адамның жұмыс орнындағы жұмысын ұйымдастыруға көмектеседі. Бірақ кеңірек контексте бұл өмірдің әр түрлі салаларында және жеке салаларында өнімділік, сондай-ақ кез-келген ұйымның тиімділігін арттырады.

Қазіргі таңда көптеген әртүрлі сұрақтарды шеше аламыз - жұмысшы ретінде де, персона ретінде де. Адам миына ақпараттың мұндай көлемімен мұқият бақылаусыз оны жеңу қиын.

1.4 Жүйені модельдеу

1.4.1 ER диаграммасы



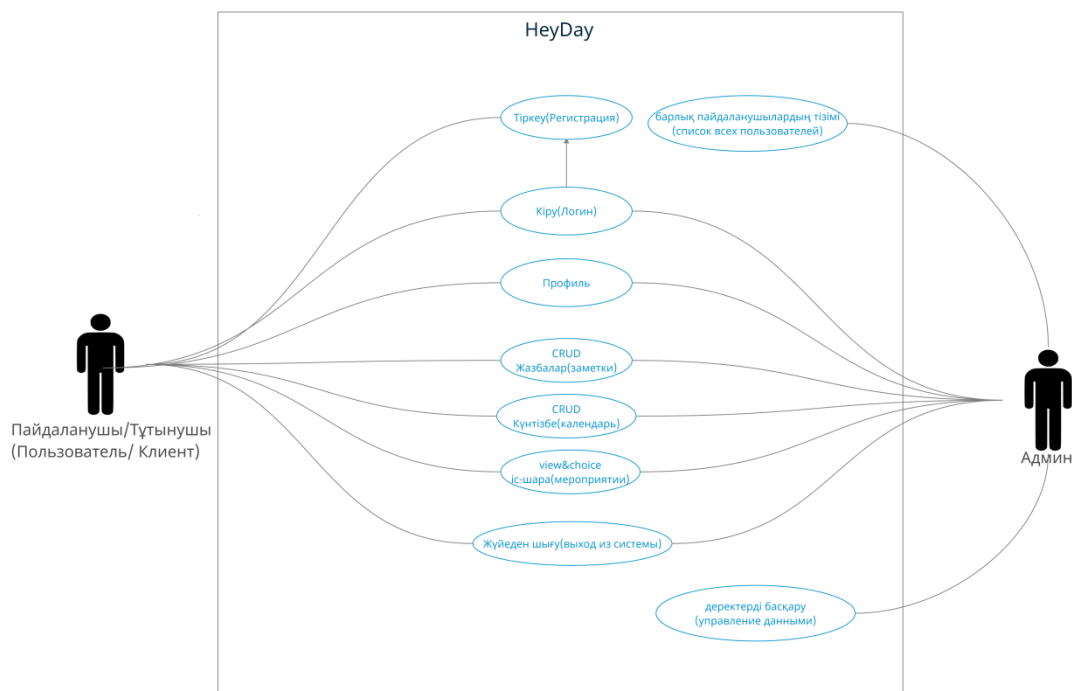
1.1-сурет – «ER диаграммасы»

Нысан байланысының диаграммасы (ER) - бұл адамдар, объектілер немесе ұғымдар сияқты "нысандар" жүйеде бір-бірімен қалай байланысты екенін көрсететін схема түрі. ER диаграммалары көбінесе бағдарламалық жасақтама, бизнес-Ақпараттық жүйелер, білім беру және зерттеу саласындағы реляциялық мәліметтер базасын жобалау немесе жөндеу үшін қолданылады. Herbs немесе ER модельдері ретінде де белгілі, олар нысандардың,

қатынастардың және олардың атрибуттарының өзара байланысын көрсету үшін тіктөртбұрыштар, ромбтар, аналық бездер және байланыстырушы сызықтар сияқты белгілі бір таңбалар жиынтығын пайдаланады. Олар грамматикалық құрылымды көрсетеді, зат есімдер мен етістіктер түріндегі қатынастар[1].

Ең басты кесте ол пайдаланушы кестесі. Осы кесте арқылы қалған коллекциядағы деректер байланысады. Пайдаланушы кестеде қай түрмен тіркелгенін, поштасын, тіркелген уақыты мен uid сақталады. Uid системдік түрде өзі береді. Events коллекцияда күнтізбе деректері сақталады. Users кестесімен байланысуы uid арқылы. Дәл солай Notes коллекциясыда байланысады. Events коллекцияда уақыт пен мәтіні сақталады. Notes коллекцияда тақырыбы, ескертпе мәтіні, жазылған уақыты мен бас бетінде қай түспен боялған индексі сақталады.

1.4.2 Use Case диаграммасы



1.2-сурет – «Use case диаграммасы»

Use Case қатысушылардың өзара әрекеттесу сценарийін сипаттайды. Диаграммада пайдаланушы мен жүйе туралы негізгісін сипаттайды.

Көп жағдайда бұл диаграмманы қолдануға болады. Оқиғалардың негізгі және баламалы ағыны арқылы тармақталу талабы сипатталғанда әзірлеушілер үшін өте ыңғайлы. Кім, қашан және не себеп, оның нәтижесі қандай екені түсінікті. Сонымен қатар, бұл тұтынушыларға ыңғайлы. Тұтынушы дәл осы күтіп тұрғанын дер кезінде растай алады немесе оны түзетеді. Сондықтан бұл

көп жағдайда қажет. Егер сізге жоғары сапалы, талаптардың толық сипаттамасы қажет болса, пайдалану жағдайлары бұған тамаша көмектеседі. Мәліметтер үлгісін, интерфейс сипаттамасын, басқа жүйелермен интеграцияны және пайдалану жағдайларын қамтитын талаптардың спецификациясы әзірлеу және қолдау үшін өте жақсы нұсқа болып табылатын осындай жүйелер бар. Жүйені қолдау үшін. Қатені анықтау үшін қай қадамда қателік кеткенін анықтауға көмектеседі.

1.2-суретте көрсетілгендей пайдаланушы мобильді қосымшаға тіркеу немесе кіре алады. Мобильді қосымшаға кіріп, жазбалар, секундомер мен күнтізбені қолданыла алады. Админ барлық тіркелген пайдаланушыларды көріп, сол деректертермен жұмыс жасай алады. Сонымен қатар бар деректерді көре алады.

1.5 UML диаграммасы

UML өзінің схемалық белгілері үшін танымал. UML бағдарламалық және бағдарламалық емес жүйелердің компоненттерін визуализациялау, нақтылау, құру және құжаттау үшін жасалғанын бәріміз білеміз. Сондықтан визуализация түсіну және есте сақтау қажет маңызды бөлік болып табылады.

UML белгілері модельдеудің маңызды элементтері болып табылады. Толық және мағыналы модель құру үшін белгілерді тиімді және дұрыс пайдалану өте маңызды. Егер оның мақсаты дұрыс көрсетілмесе, Модель пайдасыз болады.

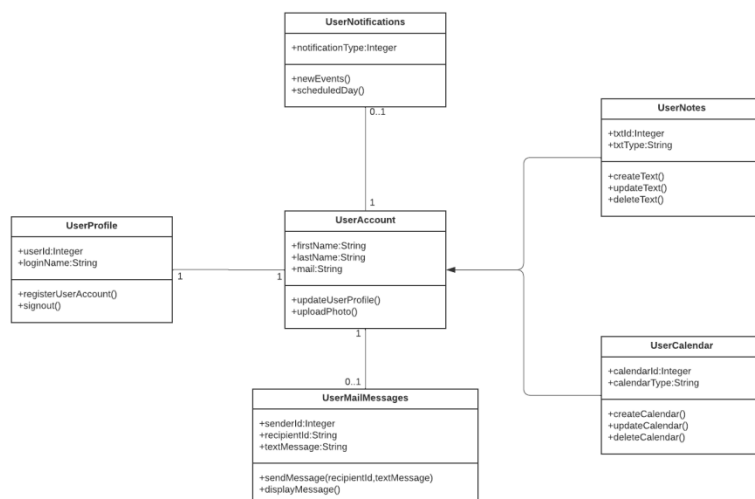
Сондықтан белгілерді зерттеуді басынан бастап атап өту керек. Заттар мен қатынастар үшін әртүрлі белгілер бар. UML диаграммалары заттар мен қатынастардың белгілерін қолдана отырып жасалады. Кеңейту-бұл UML-ді күшті және икемді ететін тағы бір маңызды мүмкіндік[2].

1.5.1 Класс диаграммасы

Класс диаграммалары-UML-де қолданылатын ең көп таралған диаграммалар. Класс диаграммасы сыныптардан, интерфейстерден, ассоциациялардан және бірлескен жұмыстардан тұрады. Класс диаграммалары негізінен статикалық жүйенің объектіге бағытталған көрінісі болып табылады.

Белсенді класс жүйенің параллелизмін көрсету үшін сынып диаграммасында қолданылады.

Класс диаграммасы жүйенің объект бағдарын білдіреді. Демек, ол әдетте даму мақсаттары үшін қолданылады. Бұл жүйені құру кезінде ең көп қолданылатын диаграмма[3].



1.3-сурет – «Берілген жұмыстың UML Класс диаграммасы»

1.3-суретте uml класс диаграммасында пайдаланушының тіркелген аккаунтынан хабарландыруын сипатталған. Күнтізбе мен ескертпеге жазылған уақытына қарап мобильді құралға хабарландыру келеді.

1.6 Технологиялар бөлімі

1.6.1 Android studio



1.4-сурет – «Android Studio логотипі»

Android studio - Android қолданбаларын әзірлеуге арналған ресми интеграцияланған әзірлеу ортасы (IDE). Android Studio Android қолданбаларын жасау кезінде өнімділікті арттыратын қосымша мүмкіндіктерді ұсынады.

Android Studio 2013 жылдың 16 мамырында Google I/O конференциясында Android қолданбасын әзірлеуге арналған ресми IDE ретінде жарияланды. Ерте қол жеткізуді алдын ала қарау 2013 жылдың мамырында 0.1 нұсқасымен басталды. Бірінші тұрақты құрастыру нұсқасы 2014 жылдың желтоқсанында 1.0 нұсқасынан бастап шығарылды.

Android Studio мүмкіндіктері:

- оның Gradle негізіндегі икемді құрастыру жүйесі бар;
- оның қолданбаны сынауға арналған жылдам және мүмкіндіктері мол эмуляторы бар;
- android Studio-да барлық Android құрылғылары үшін әзірлей алатын бірыңғай орта бар;
- қолданбаны қайта іске қоспай-ақ, біздің іске қосылған қолданбаның ресурстық кодына өзгертулерді қолданыңыз;
- android Studio кең ауқымды тестілеу құралдары мен құрылымдарын ұсынады;
- ол C++ және NDK тілдерін қолдайды;
- ол Google Cloud Platform үшін жергілікті қолдауды қамтамасыз етеді;
- бұл Google Cloud Messaging және App Engine интеграциясын жеңілдетеді.

Android Studio бағдарламасындағы әрбір жобада бастапқы код файлдары мен ресурс файлдары бар бір немесе бірнеше модульдер бар. Модуль түрлеріне мыналар жатады:

- android қолданбалы модульдері;
- кітапхана модульдері;
- google App Engine модульдері[4].

1.6.2 Бағдарламау тілі



1.5-сурет – «Dart программалау тілінің логотипі»

Dart — әзірлеушілерге статикалық типті талдау арқылы объектіге бағытталған тілді пайдалануға мүмкіндік беру мақсатында Google әзірлеген ашық бастапқы тіл. Бірінші тұрақты шығарылымнан бері Dart тілдің өзінде де, оның негізгі мақсаттарында да көп нәрсені өзгертті. 2.0 нұсқасында Dart типтік

жүйесі қосымшадан статикалық жүйеге көшті және оның құрылған күнінен бастап Flutter (бұл туралы кейінірек түсіндіреміз) тілдің негізгі мақсаты болды.

Көптеген тілдерден айырмашылығы, Dart әзірлеушілер үшін әзірлеу процесін барынша жеңіл және жылдам ету мақсатында жасалған. Сонымен, ол өзінің пакет менеджері, әртүрлі компиляторлар/транспилиаторлар, талдаушы және пішімдеуші сияқты кіріктірілген құралдардың өте кең жиынтығымен келеді. Бұған қоса, Dart VM және Just-in-Time құрастырулары кодты дереу өзгертуге мүмкіндік береді.

Іске қосылғаннан кейін кодты ана тілінде құрастыруға болады, сондықтан іске қосу үшін арнайы орта қажет емес. Веб-әзірлеу жағдайында Dart JavaScript-ке тасымалданады.

Синтаксис тұрғысынан Dart JavaScript, Java және C++ сияқты тілдерге өте ұқсас, сондықтан осы тілдердің бірімен Dart тілін үйрену бірнеше сағатты қажет етеді.

Сонымен қатар, Dart асинхронияға үлкен қолдау көрсетеді және генераторлармен және қайталанатын құрылғылармен жұмыс істеу өте оңай.

Dart - бұл жалпы мақсаттағы тіл және сіз оны кез келген нәрсе үшін пайдалана аласыз:

- веб қолданбаларында көркем кітапхананы пайдалану: html және Dart кодын JavaScript-ке түрлендіру үшін транспилер немесе AngularDart сияқты фреймворктарды пайдалану;
- `art:http` және `art:io` кітапханаларын пайдаланатын серверлерде. Сондай-ақ, `Aqueduct` сияқты бірнеше рамкаларды пайдалануға болады;
- консольдық қолданбаларда;
- мобильді қолданбаларда Flutter арқасында[5].

1.6.3 Flutter framework



1.6-сурет – «Flutter фреймворкінің логотипі»

Flutter — әзірлеуші бір кодтық базадан жергілікті Android және iOS қолданбаларын құру үшін пайдалана алатын ашық бастапқы мобильді SDK.

Flutter 2015 жылы Google оны ұсынғаннан бері жұмыс істейді және 2018 жылдың желтоқсанында ресми іске қосылғанға дейін бета-сынақта қалды. Содан бері Флуттердің айналасындағы хайп күшейе түсті.

Flutter қазір GitHub жұлдыздарына негізделген бағдарламалық қамтамасыз етудің үздік 11 репозиторийінде. Сонымен қатар, біз қолданбалар дүкендерінде жарияланған мыңдаған Flutter қолданбаларын көрдік. Ең көрнекті мысалдардың бірі - Alibaba командасы жасаған Xianyu қолданбасы, оны 50 миллионнан астам адам пайдаланады.

Flutter-тің негізгі идеясы - виджеттерді пайдалану. Әртүрлі виджеттерді біріктіру арқылы әзірлеушілер бүкіл пайдаланушы интерфейсін жасай алады. Бұл виджеттердің әрқайсысы құрылымдық элементті (мысалы, түйме немесе мәзір), стильдік элементті (қаріп немесе түс схемасы сияқты), орналасу аспектісін (мысалы, толтыру) және т.б. анықтайды.

Flutter жаңа кросс-платформалық құрылым болғанымен, көбірек компаниялар Xamarin, Cordova және React Native сияқты фреймворктардан гөрі Flutter-ті таңдайды.

Әзірлеуші топтардың Flutter-ті таңдауының негізгі себептері:

- жетілдірілген өнімділік. iOS және Android үшін бірдей код базасын пайдалану уақыт пен ресурстарды үнемдейді. Флуттердің жеке виджеттері де ОС әртүрлі нұсқаларымен үйлесімділік мәселелерінің болмауын қамтамасыз етіп, сынақ уақытын азайтады;
- үйрету оңай. Flutter әзірлеушілерге OEM виджеттеріне қол жеткізбей немесе көп кодты пайдаланбай-ақ өздерінің мобильді қолданбаларын жасауға мүмкіндік береді. Бұл Flutter-тің ерекше тартымды пайдаланушы интерфейсіне қоса, мобильді қосымшаны құру процесін айтарлықтай жеңілдетеді;
- тамаша өнімділік. Пайдаланушылар Flutter қолданбасы мен жергілікті мобильді қолданбаны ажырату қиын екенін айтады;
- үнемді. Бірдей код базасын пайдаланып iOS және Android қолданбаларын құру негізінен біреуінің бағасына екі қолданбаны құру болып табылады;
- әртүрлі IDE-де қол жетімді. Әзірлеушілер Flutter кодын өңдеу үшін Android Studio және VS Code арасында еркін таңдай алады;
- тамаша құжаттама және қауымдастық. Flutter қолданбасында қарапайым қолдану жағдайлары бар кең құжаттамасының арқасында сұрақтарыңызға жауап беретін көптеген тамаша ресурстар бар. Flutter пайдаланушылары идеялармен бөлісу үшін Flutter Community және Flutter Awesome сияқты қауымдастық хабтарын да пайдалана алады.

Flutter құрылымы мобильді қолданбаларды әзірлеушілер үшін де, бизнес үшін де қызықты мүмкіндіктер береді. Бірдей код базасы және өте ыңғайлы интерфейсін бар iOS және Android үшін қолданбаларды жасау мобильді қолданбаларды әзірлеуді жылдам және үнемді етеді[6].

1.6.4 Figma



1.7-сурет – «Figma сервисінің логотипі»

Figma - кез келген нәрсені жасауға болатын қуатты дизайн құралы: веб-сайттар, қолданбалар, логотиптер және т.б. Фигма басқа графикалық өңдеу құралдарынан ерекшеленеді. Негізінен ол браузерде дұрыс жұмыс істейді. Бұл бірнеше лицензияларды сатып алуды немесе бағдарламалық құралды орнатуды қажет етпей-ақ кез келген компьютерден немесе платформадан дизайнға қол жеткізуге және жобалауды бастауға болатынын білдіреді. Негізінде, Figma пайдаланушы интерфейстерін жобалау және прототиптеу үшін жасалған құрал болып табылады.

1.6.5 Firebase



1.8-сурет – «Firebase сервисінің логотипі»

Firebase — қолданбаңызды жасауға, жақсартуға және дамытуға көмектесетін Google мобильді қолданбаларды әзірлеу платформасы.

Firebase – «қолданбаңызды құруға, жақсартуға және дамытуға» арналған құралдар жинағы және ол ұсынатын құралдар әзірлеушілер әдетте өздері құрастыруға тура келетін, бірақ шынымен жасағысы келмейтін қызметтердің көпшілігін қамтиды, өйткені олар назар аударған дұрыс. нақты қолдану. Бұған аналитика, аутентификация, дерекқорлар, конфигурация, файлдарды сақтау,

push хабарлары сияқты нәрселер кіреді және тізім жалғасады. Қызметтер бұлтта орналастырылады және әзірлеушінің аз немесе мүлдем күш-жігерінсіз масштабталады.

Бұл әдетте алдыңғы және артқы жағындағы бағдарламалық құралды жазуды қамтитын дәстүрлі қолданбаларды әзірлеуден ерекшеленеді. Сыртқы код қарапайым интерфейспен қамтамасыз етілген API соңғы нүктелерін шақырады, ал ішкі код іс жүзінде барлық жұмысты жасайды. Дегенмен, Firebase өнімдерінде дәстүрлі сервер жағы айналып өтеді, жұмыс клиентте орындалады. Осы өнімдердің әрқайсысына әкімшілік қол жеткізу Firebase консолі арқылы қамтамасыз етіледі.

Firebase өнімдері көмектесетін қолданбалардың түрлеріне ешқандай шектеулер жоқ. Оны пайдалануға болатын платформаларда ғана шектеулер бар. iOS және Android - Firebase SDK үшін негізгі мақсаттар және веб-қосымшаларға, Flutter, Unity және C++ қолдауы кеңейіп келеді.

Аутентификация – логин және пайдаланушы идентификациясы. Нақты уақыттағы деректер базасы - нақты уақыттағы, бұлттық деректер базасы, NoSQL бұлттық дерекқоры. Firestore - нақты уақыт режимінде, бұлттық деректер базасы, NoSQL деректер базасы.

Firebase аутентификациясы пайдаланушыларыңыздың жүйеге кіріп, өздерін анықтауын қамтамасыз етеді. Бұл өнім кейбір басқа өнімдерді дұрыс конфигурациялау үшін қажет, әсіресе деректерге қол жеткізуді әр пайдаланушы негізінде шектеу қажет болса.

Firebase Realtime Database және Cloud Firestore дерекқор қызметтерін ұсынады. Олардың екеуін де «нақты уақытта бұлтта орналастырылған NoSQL дерекқорлары» деп атады. Олардың жеке күшті және әлсіз жақтары бар және олардың қайсысы сіздің қажеттіліктеріңізге сәйкес келетінін білу үшін біраз зерттеулер жүргізу қажет болуы мүмкін. Cloud Firestore-дан бастадым, себебі ол менім қажетті құралдары бар. Firestore техникалық тұрғыдан Firebase өнімі емес, Google Cloud өнімі болып табылады. Firebase деректерге тікелей қол жеткізуді қосу үшін мобильді қолданбада пайдалану үшін SDK қосады, бұл жағымсыз орта құрамдас бөлігінің қажеттілігін болдырмайды. Google Cloud-пен ұқсас қарым-қатынасы бар басқа өнімдер осында берілген, мен оларды да ескеремін.

Бұл дерекқорлардың шын мәнінде ерекшелігі, олар дерекқорда өзгерген кезде деректер жаңартуларын «нақты уақытта» қамтамасыз етеді. Қолданбаңыз пайдаланғысы келетін деректер орнында "тыңдаушыны" орнату үшін клиент SDK пайдаланып және өзгеріс байқалған сайын тыңдаушы сол деректермен қайта шақырылады. Бұл қызықты деректерді сұраусыз қолданбаның дисплейін жаңартуға мүмкіндік береді.

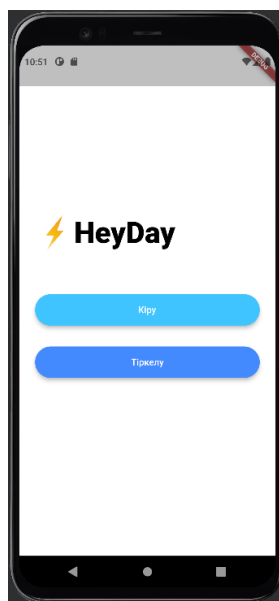
Firebase Hosting – қауіпсіз жаһандық CDN (контентті жеткізу желісі) веб-хостинг қызметі. Бұл пайдаланушыларға жақын серверлерді пайдаланып, статикалық мазмұнды (HTML, CSS, JS, суреттер) жылдам жеткізу үшін өте жақсы. Сіз оны теңшелетін доменмен немесе доменсіз жылдам орнатуға

болады, сонымен қатар сізге ешнәрсе төлемейтін дайындалған SSL сертификаты бар.

Firebase үшін ML жинағы машиналық оқыту туралы ештеңе білмей-ақ Google-дың бай машиналық оқыту тәжірибесін пайдалануға мүмкіндік береді. Бұл мен үшін өте жақсы, өйткені мен ML туралы ештеңе білмеймін! Бірақ ML Kit қолданбасынан алатын нәрсе - мәтін, беттер және бағдарлар сияқты құрылғының камерасы түсіріп жатқан нәрселерді тану мүмкіндігі. Және ол менің мобильді құрылғымда өте шектеулі өңдеу қуатымен жұмыс істей алады[7].

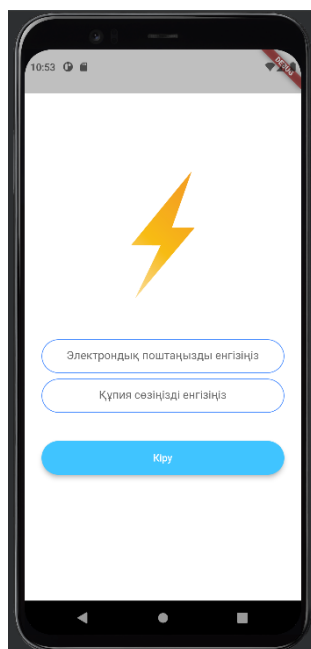
2 Жобаның құрылымы

2.1 Аутентфикация бөлімі



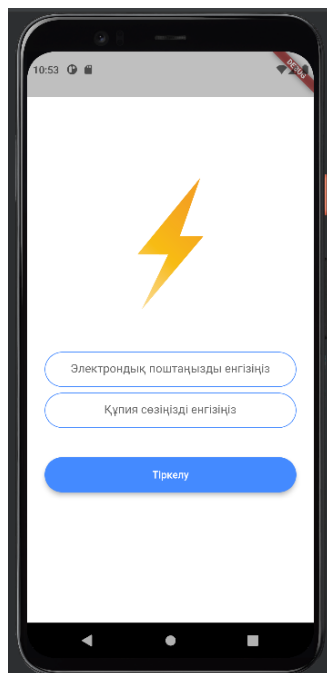
2.1-сурет – «Бастапқы беті»

Бастапқы бетте бізге тіркелуге немесе кіруге нұсқауды көрсетіп тұр.



2.2-сурет – «Кіру беті»

Кіру бетінде электронды поштаны жазып, құпия сөзді енгізу керек. Содан кейін деректер базасындағы деректер тексеріп, барлық деректер дұрыс болса мобильді қосымшаға кіруге болады.

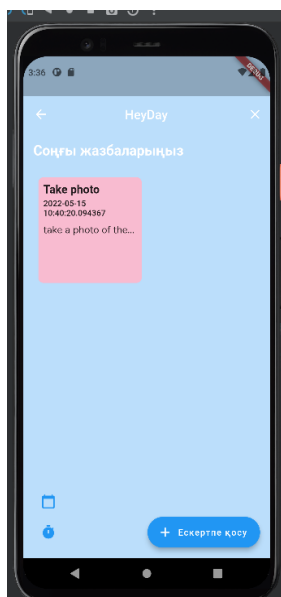


2.3-сурет – «Тіркелу беті»

Тіркелу бетінде жаңа пайдаланушы ретінде кіріп, кейін тіркеу беті арқылы мобильді қосымшаға кіруге болады.

2.2 Қолданушыдың басты бет көрінісі

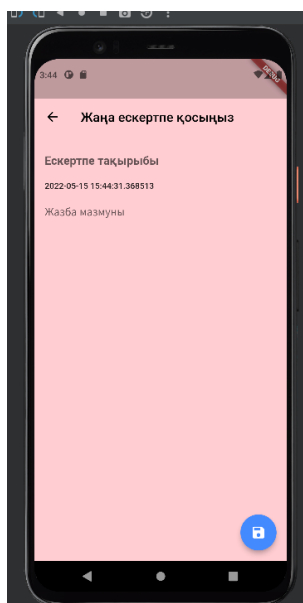
Басты бетінде барлық жазбаларды көруге болады. Осы жазбаларға қарап, керекті мәліметтер алуға немесе оны өзгертуге болады. Басты бетінде үш батырма бар. Әр батырма керекті басқа бетке көшіреді. Пайдаланушы мобильді қосымшадан шыққысы келсе керекті устінгі оң жақтағы батырмасына басып шыға алады.



2.4-сурет – «Басты беті»

Басты бетте деректер қорда сақталған ескертпелер сақталады. Сонымен қатар секундомер, күнтізбеге және тағыда ескертпе қосу үшін батырмалары бар. Ескертпеде басты тақырыбы, уақытысы және мәтіні сақталады. Соңғы жазбараларын түстеріде деректер қорда сақталады.

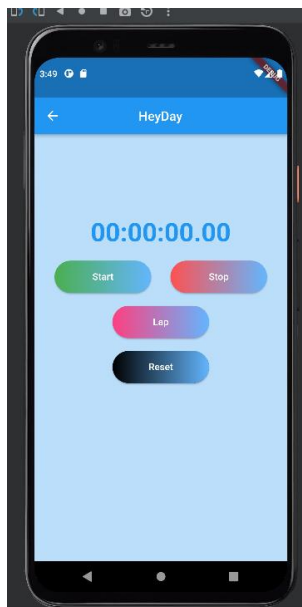
2.3 Ескертпе қосу беті



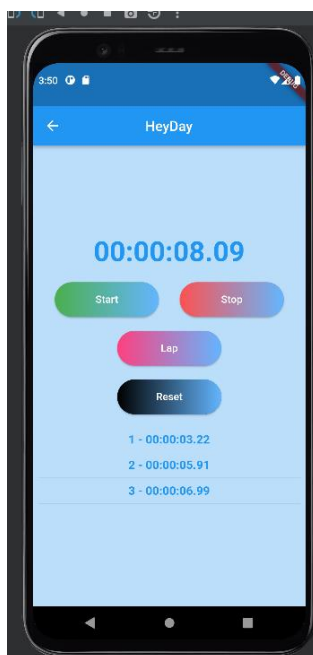
2.5-сурет – «Ескертпе қосу беті»

Ескертпе бетінде бас бетінде қай түспен көрініп тұратын түсті фонын бояйды. Сонымен қатар уақыты көрініп тұрады.

2.4 Секундомер беті



2.6-сурет – «Секундомер беті»



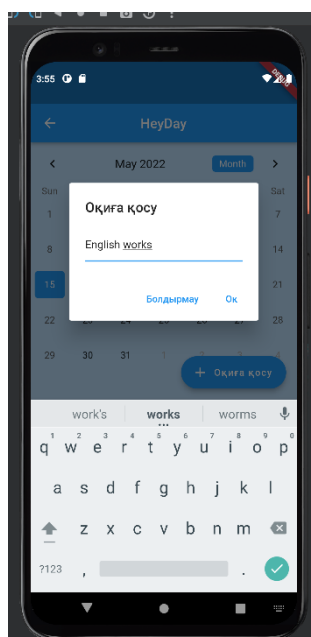
2.7-сурет – «Секундомер беті қосылғанда»

2.7-суретте уақытты қадалауға көмектеседі. Осында уақыттын интервалдарын секундтың ең жақын бөлігіне дейін өлшеуге қабілетті.

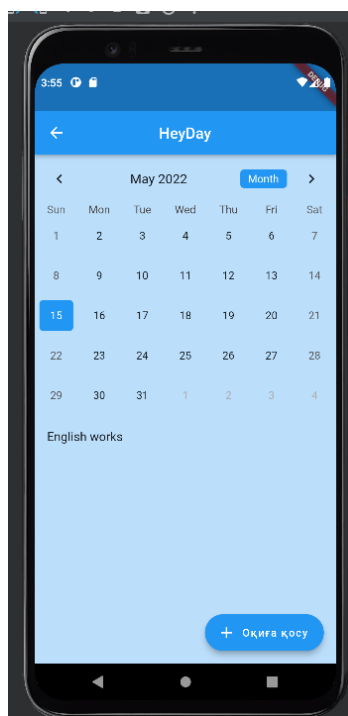
2.5 Күнтізбе беті



2.8-сурет – «Күнтізбе беті»



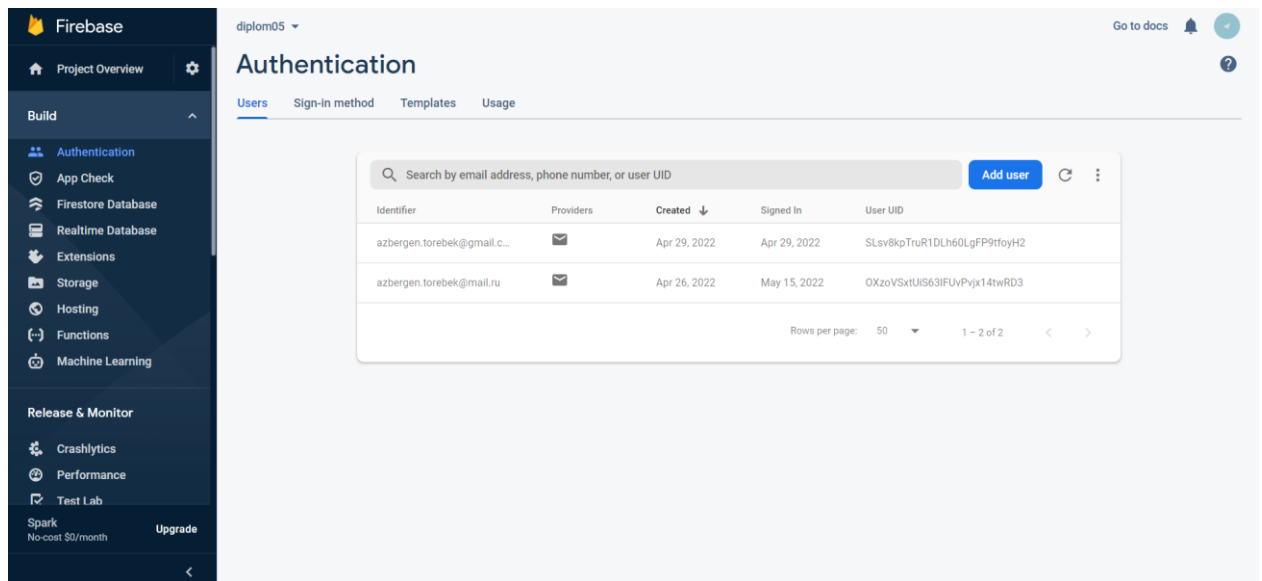
2.9-сурет – «Күнтізбе бетінде оқиғаны қосу»



2.10-сурет – «Күнтізбе бетінде оқиғаны көрінуі»

3 Firebase деректермен жұмыс бөлімі

3.1 Firebase аутенфикация бөлімі



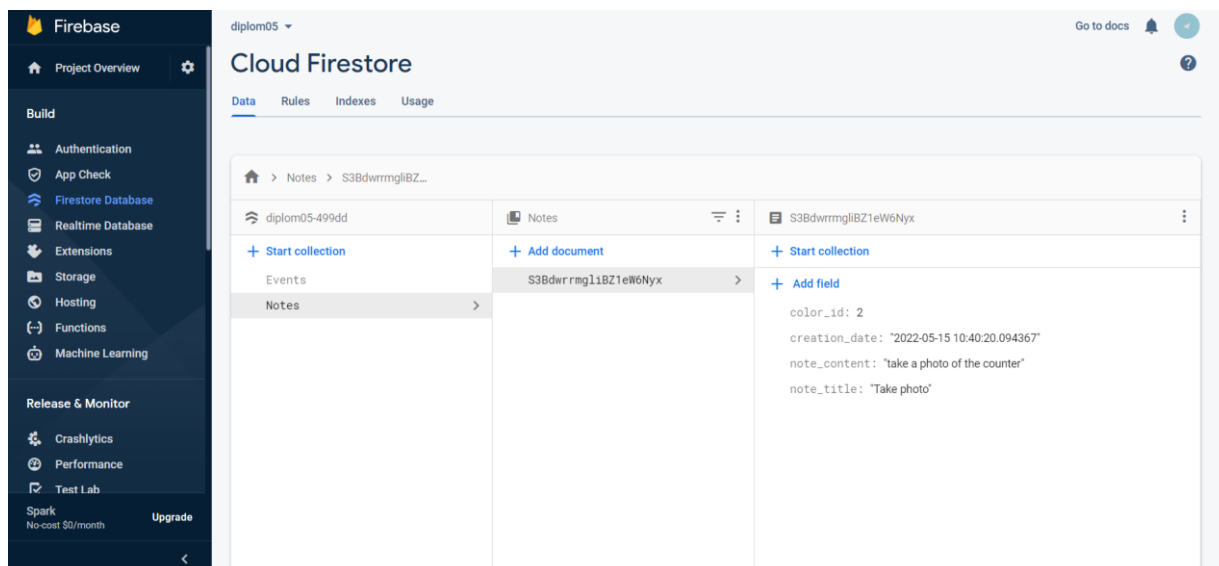
3.1-сурет – «Firebase аутенфикация бөлімі»

Аутенфикация бөлімінде неше пайдаланушылар тіркелегенін және кім екенін көре аламыз. Ол үшін Authentication батырмасына басып көруге болады. Сонымен қатар кім неше рет келгенін көріп, қадалауға мүмкіндік бар. Оны 3.1-суретте көруге болады. Кіру методын таңдап, қалағанына ауыстыруға болады.

3.2 Firebase пайдаланушының деректер қорымен жұмыс бөлімі

3.2.1 Ескертпе деректері

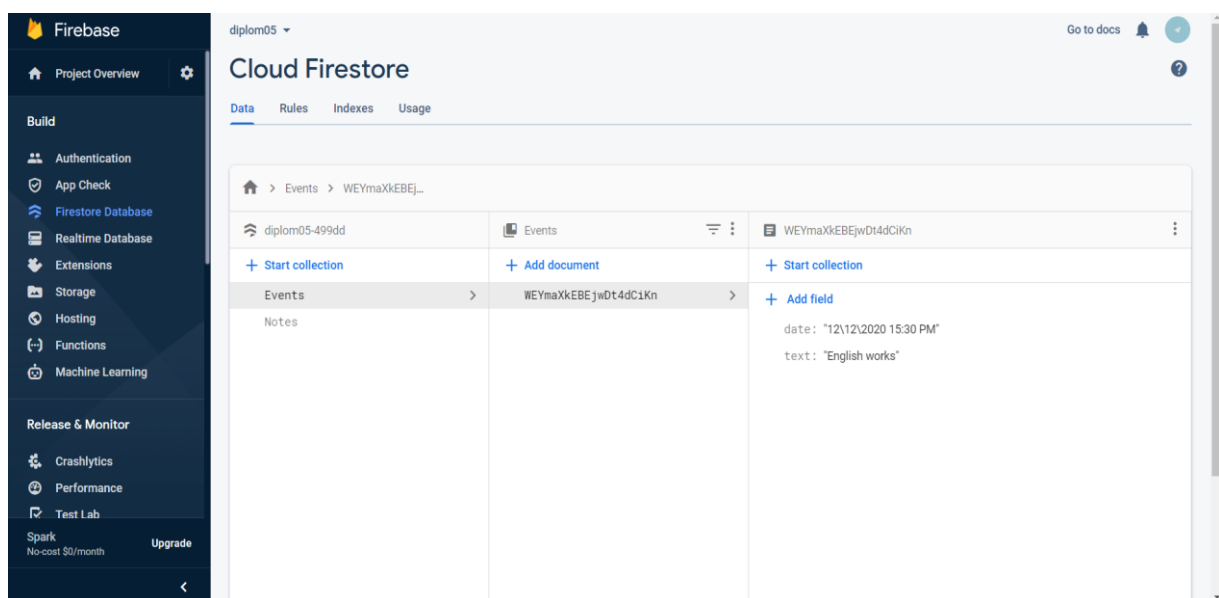
Ескертпе деректер қолданбасында жалпы уақытты басқаруды жақсартуға арналған бірқатар ресурстар бар. Олар уақытты басқарудағы жалпы қиындықтарға практикалық шешімдерді, сондай-ақ негізгі әдеттерді жақсы жаққа өзгерту жолдарын ұсынады.



3.2-сурет – «Firebase ескертпе деректері»

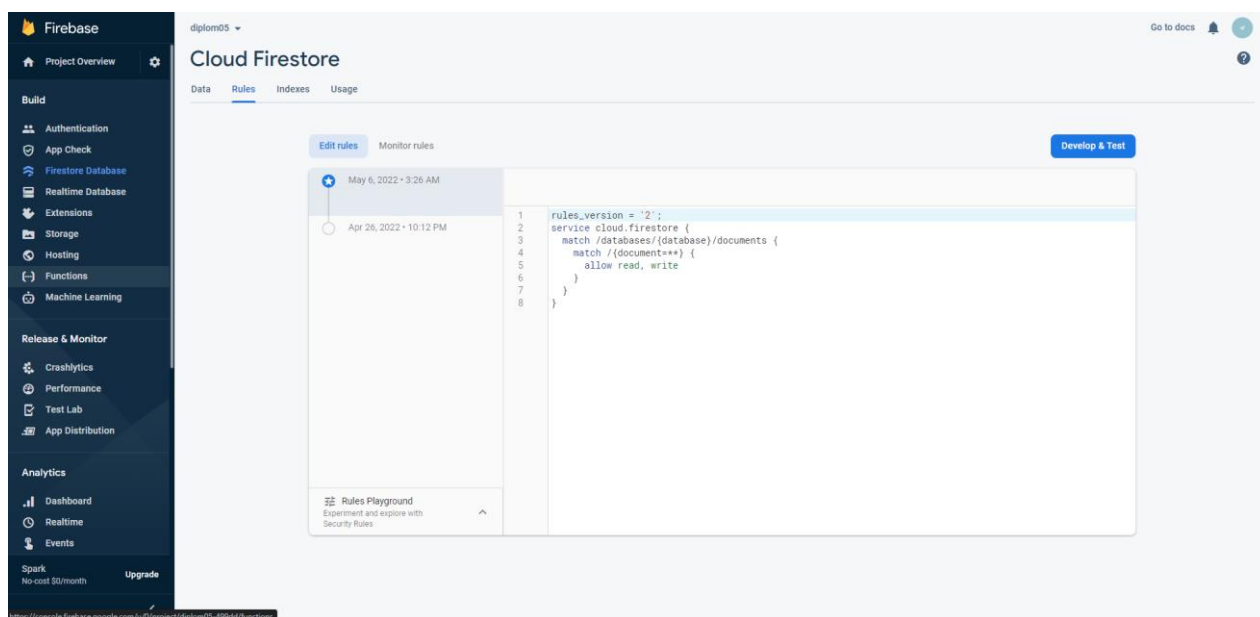
Firebase деректері Nosql қолданылады. Ескертпе деректері Notes коллекцияда сақталынады.

3.2.2 Күнтізбе ескертпелері



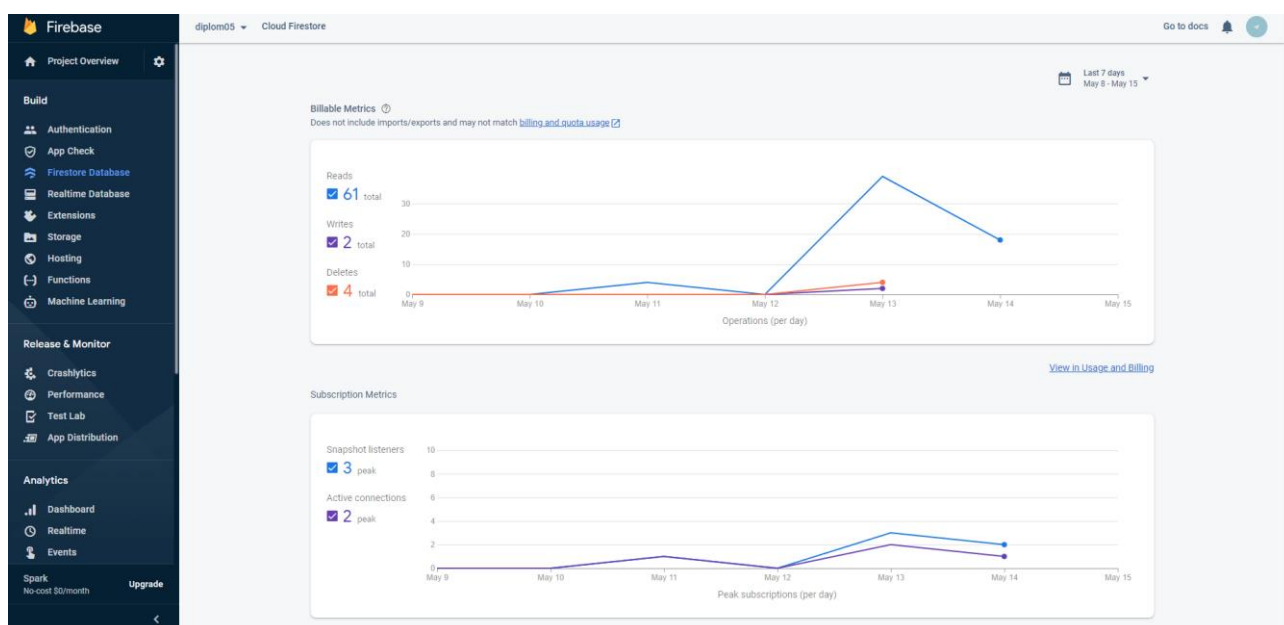
3.3-сурет – «Firebase күнтізбе деректері»

3.2.3 Ескертпе ережесі



3.4-сурет – «Firebase деректері ережесі»

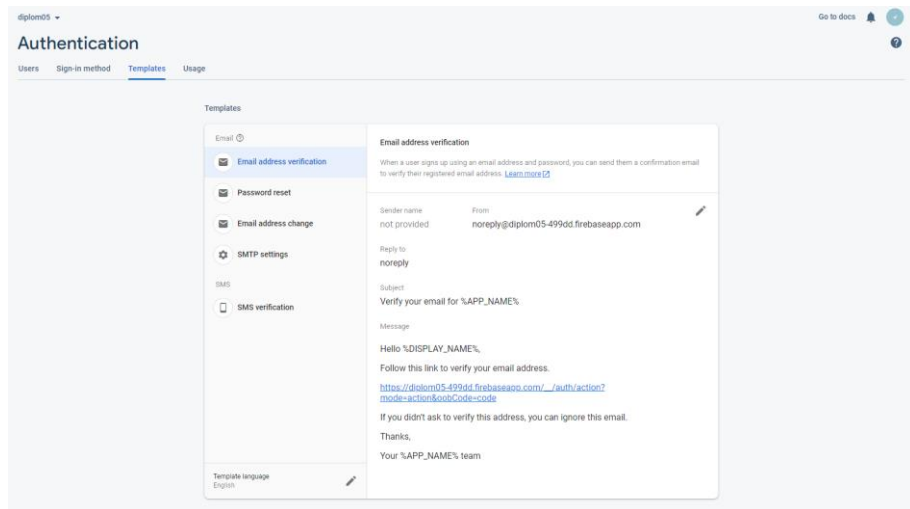
3.2.4 Ескертпенің қолдануы



3.5-сурет – «Firebase деректері қолдану графиктері»

4 Пошта үлгілері

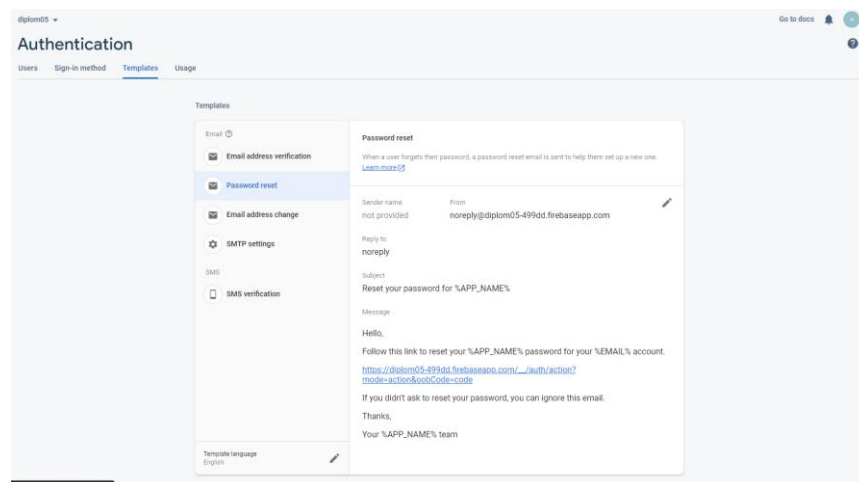
4.1 Электрондық пошта мекенжайын тексеру



4.1-сурет – «Электрондық пошта мекенжайын тексеру структурасы»

4.1-суретте пайдаланушы электрондық пошта мекенжайы мен құпия сөзді пайдаланып тіркелген кезде, олардың тіркелген электрондық пошта мекенжайын растау үшін оларға растау электрондық поштасын жіберуге болатын хабарландырудың сөйлемдері.

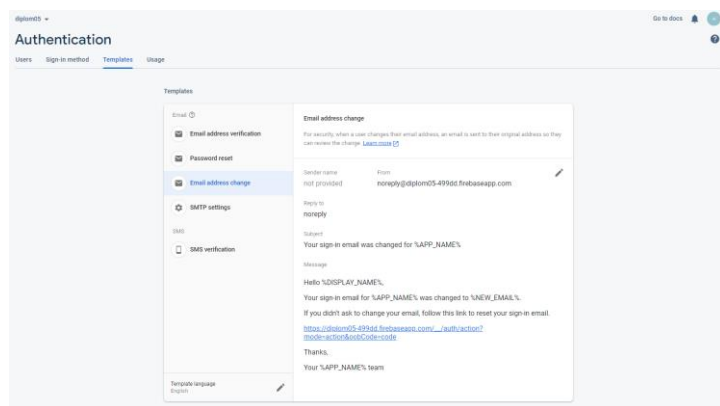
4.2 Құпия сөзді қалпына келтіру



4.2-сурет – «Құпия сөзді қалпына келтіру структурасы»

Пайдаланушы құпия сөзін ұмытып қалса, жаңасын орнатуға көмектесу үшін құпия сөзді қалпына келтіру электрондық поштасы жіберіледі. Осында хабарландырудың толық сөйлемін көріге болады.

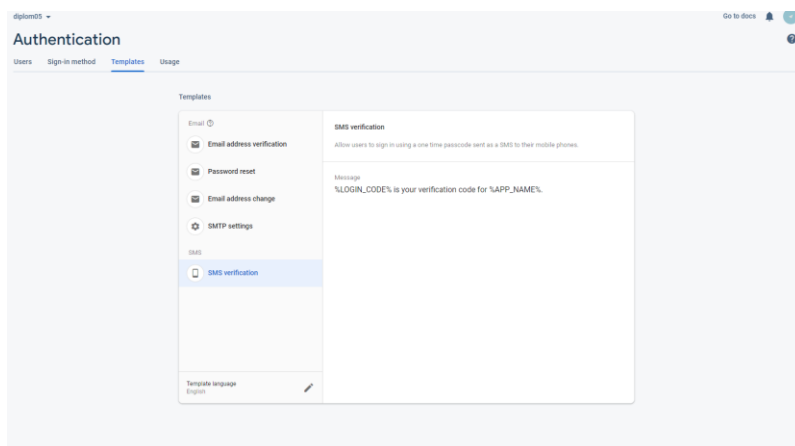
4.3 Электрондық пошта мекенжайын өзгерту



4.3-сурет – «Электрондық пошта мекенжайын өзгерту структурасы»

Қауіпсіздік мақсатында пайдаланушы электрондық пошта мекенжайын өзгерткен кезде, өзгертуді қарап шығуы үшін олардың бастапқы мекенжайына электрондық хат жіберіледі.

4.4 SMS растау



4.4-сурет – «SMS растау структурасы»

Пайдаланушыларға ұялы телефондарына SMS ретінде жіберілген бір реттік рұқсат кодын пайдаланып жүйеге кіруге рұқсаты.

ҚОРЫТЫНДЫ

Дипломдық жұмыстың нәтижесінде Dart программалау тілін қолданып, уақытты тиімді пайдалануға арналған мобильді қосымша жасалды. Орындалу барысында күнтізбеде қолданушыға керек күнді тандап, өзіне арналған тапсырмаларды, өміріндегі оқиғаларды жазып қоя алады. Жәнеде өзінің орындануына кеткен уақытын секундомер арқылы бақылай алады.

Мобильді қосымша Flutter фреймворкінің функциясы арқылы кроссплатформалы болып табылады. Яғни IOS және Android құрылғыларында оп-оңай қолдануға болады.

Қазіргі кезде адам үшін уақытты басқару ең маңызды істердің ағыннан бөліп, оларды орындау үшін қажет. Көзге жағымды интерфейстің арқасында қосымшамен пайдаланушы тез жұмыс жасай алады. Уақытты тиімді қолдану, тек жоғары басымдықта тұрған жұмыстарды. тапсырмаларды орындау қолданушы өмірін жеңілдете түседі.

Болашақта көптеген жаңа функционалдарды қосу жоспарланып отыр. Сонын ішінде достарды қабылдау, басқада адамдарға достық жіберу, өз күн кестесімен достарымен бөлісуге болады. Корпоративтік пошта арқылы компания ішіндегі жұмысшылардың кестесін көре алу, жасалған жұмыстарын көре алады. Сонын арқасында жұмыс уақытынды дұрыс қолдануға, түсініспеушіліктерді азайтады.

ПАЙДАЛАНЫЛҖАН ӘДЭБИЕТТЕР ТІЗІМІ

- 1 <https://www.lucidchart.com/pages/er-diagrams>
- 2 https://www.tutorialspoint.com/uml/uml_basic_notations.htm
- 3 https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm#:~:text=Class%20diagrams%20are%20the%20most,the%20concurrency%20of%20the%20system.
- 4 https://medium.com/@dsc_3204/android-studio-android-studio-is-the-official-integrated-development-environment-ide-for-809d896df7d
- 5 <https://inlab.fib.upc.edu/en/blog/what-dart-programming-language#:~:text=What%20is%20Dart%3F,and%20in%20its%20primary%20goals.>
- 6 <https://blog.codemagic.io/what-is-flutter-benefits-and-limitations/>
- 7 <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- 8 <https://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037e>
- 9 <https://fonts.google.com/>
- 10 <https://docs.flutter.dev/>
- 11 <https://dribbble.com/>
- 12 <https://api.flutter.dev/index.html>
- 13 <https://pub.dev/>
- 14 <https://www.materialpalette.com/>
- 15 <https://docs-flutter-io.firebaseio.com/>
- 16 <https://apptimize.com/docs/installation/flutter-installation.html>
- 17 <https://docs.flutterflow.io/>
- 18 <https://docs.smartlook.com/docs/sdk/flutter?platform=react>
- 19 <https://docs.sentry.io/platforms/flutter/>
- 20 <https://flutterdocs.ru/>

А қосымшасы (міндетті)

Техникалық тапсырма

Бос күндерінізді өткізуге көмектесетін мобильді қосымшаны әзірлеу техникалық тапсырмасы

А.1 Жалпы сипаттама

Мобильді қосымшаны қолдана отырып, барлық маңызды жаңалықтар мен еске салғыштарды жазбаларды қолдана отырып жазуға болады. Сонымен қатар, өмірдің әртүрлі аспектілері үшін жеке және жұмыс күнтізбесін жасаңыз. Әр түрлі іс-шараларды бақылауға болады.

А.2 Мақсаты

Уақытты басқару уақытты тиімді пайдалану құралы ретінде ғана емес (классикалық тәсілге тән), сонымен қатар ұйымды дамыту құралы ретінде де қолданыла алатындығы дәлелденді. Осылайша, ұйымды басқару жүйесінің мәжбүрлеу және олар құрылған уақытты басқару әдістерін еркін қолдану арасындағы қайшылық шешілуі мүмкін.

А.3 Функционалдық сипаттамаларына қойылатын талаптар

- авторизация
- ескертпе;
- CRUD ескертпе операциясы;
- күнтізбе;
- секундомер.

А қосымшасының жалғасы (міндетті)

А.4 Сенімділікке талап

Мобильдік қосымша жүйесі дұрыс және тұрақты жұмыс жасауы керек. Сол себепті сенімділік келесі талаптарға сай болуы керек :

- қолданушы яғни админ болсын тіркелген жеке беттеріне кіре алады;
- ақпарат қатаң түрде мазмұнына, өріс атауларына сәйкес енгізілуі тиіс;
- енгізілген ақпараттың тұтастығын қамтамасыз ету;
- қолданушының қате әрекеттерінен қорғауды қамтамасыз ету.

А.5 Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар

Мобильді құрылғы үйлесімді жұмыс жасауы керек. Ең минималды конфигурация :

- андроид 5 тен төмен болмауы;
- жад сыйымдылығы – 1 гб және одан жоғары.

А.6 Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар

Құрылғыны іске асыратын андроид жүйелерінің бақылауында жұмыс жасауы керек.

А.7 Бағдарламалық қамтаманың құжаттамасына қойылатын талаптар

Бағдарламалық жасақтаманың құжаттамасы мыналарды қамтиды:

- техникалық тапсырма. Онда бағдарламалық қамтамасыз етуге қойылатын талаптар, әзірлеудің қажетті кезеңдері мен мерзімдері қамтылады;
- бағдарлама мәтіні – қажетті түсініктемелері бар бағдарлама жазбалары;
- логикалық құрылымы мен жұмысы туралы ақпаратты қамтитын бағдарламаның сипаттамасы.

Б қосымшасы (міндетті)

Бағдарлама коды

1 Бастапқы бет

```
class WelcomeScreen extends StatefulWidget {
  static String id = 'welcome_screen';
  @override
  _WelcomeScreenState createState() => _WelcomeScreenState();
}

class _WelcomeScreenState extends State<WelcomeScreen>
  with SingleTickerProviderStateMixin {
  late AnimationController controller;
  late Animation animation;
  @override
  void initState() {
    super.initState();
    controller = AnimationController(
      duration: Duration(seconds: 1),
      vsync: this,
      //upperBound: 100.0
    );
    //animation = CurvedAnimation(parent: controller, curve:
    Curves.easeIn);
    animation = ColorTween(begin: Colors.blueGrey, end: Colors.white)
      .animate(controller);
    //controller.reverse(from: 1.0);
    controller.forward();
    /*animation.addListener((status) {
      if(status == AnimationStatus.completed){
        controller.reverse(from: 1.0);
      } else if(status == AnimationStatus.dismissed){
        controller.forward();
      }
    });*/
    controller.addListener(() {
      setState(() {});
    });
  }

  @override
  void dispose() {
    controller.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: animation.value,
      body: Padding(
        padding: EdgeInsets.symmetric(horizontal: 24.0),
        child: Column(
```

(міндетті)

2 Кіпу беті

```
@override
```


Б қосымшасының жалғасы (міндетті)

```
_LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _auth = FirebaseAuth.instance;
  late String email;
  late String password;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Padding(
        padding: EdgeInsets.symmetric(horizontal: 24.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: <Widget>[
            Hero(
              tag: 'logo',
              child: Container(
                height: 200.0,
                child: Image.asset('images/logo.png'),
              ),
            ),
            SizedBox(
              height: 48.0,
            ),
            TextField(
              keyboardType: TextInputType.emailAddress,
              textAlign: TextAlign.center,
              onChanged: (value) {
                email = value;
              },
              decoration:
                kTextFielDecoration.copyWith(hintText:
поштаңызды енгізіңіз'),
            ),
            SizedBox(
              height: 8.0,
            ),
            TextField(
              obscureText: true,
              textAlign: TextAlign.center,
              onChanged: (value) {
                password = value;
              },
              decoration:
                kTextFielDecoration.copyWith(hintText: 'Құпия сөзіңізді
енгізіңіз'),
            ),
            SizedBox(
              height: 24.0,
            ),
          ],
        ),
      ),
    );
  }
}
```

Б қосымшасының жалғасы (міндетті)

```
RoundedButton(  
  title: 'Kipy',  
  colour: Colors.lightBlueAccent,  
  onPressed: () async {  
    try {  
      final user = await _auth.signInWithEmailAndPassword(  
        email: email, password: password);  
      if (user != null) {  
        //Navigator.pushNamed(context, ChatScreen.id);  
        Navigator.pushNamed(context, NoteScreen.id);  
      }  
    } catch (e) {  
      print(e);  
    }  
  },  
),  
],  
,  
,  
,  
);  
}
```

3 Тіркелу беті

```
class RegistrationScreen extends StatefulWidget {  
  static String id = 'registration_screen';  
  @override  
  _RegistrationScreenState createState() => _RegistrationScreenState();  
}  
class _RegistrationScreenState extends State<RegistrationScreen> {  
  final _auth = FirebaseAuth.instance;  
  late String email;  
  late String password;  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      backgroundColor: Colors.white,  
      body: Padding(  
        padding: EdgeInsets.symmetric(horizontal: 24.0),  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          crossAxisAlignment: CrossAxisAlignment.stretch,  
          children: <Widget>[  
            Hero(  
              tag: 'logo',  
              child: Container(  
                height: 200.0,  
                child: Image.asset('images/logo.png'),  
              ),  
            ),  
            SizedBox(  
              height: 48.0,  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

Б қосымшасының жалғасы (міндетті)

```
TextField(
  keyboardType: TextInputType.emailAddress,
  textAlign: TextAlign.center,
  onChanged: (value) {
    email = value;
  },
  decoration:
    kTextFielDecoration.copyWith(hintText: 'Электрондық
поштаңызды енгізіңіз'),
),
SizedBox(
  height: 8.0,
),
TextField(
  obscureText: true,
  textAlign: TextAlign.center,
  onChanged: (value) {
    password = value;
  },
  decoration:
    kTextFielDecoration.copyWith(hintText: 'Құпия сөзіңізді
енгізіңіз'),
),
SizedBox(
  height: 24.0,
),
RoundedButton(
  title: 'Тіркелу',
  colour: Colors.blueAccent,
  onPressed: () async {
    try {
      final newUser = await
_auth.createUserWithEmailAndPassword(
        email: email, password: password);
      if (newUser != null){
        Navigator.pushNamed(context, ChatScreen.id);
      }
    }
    catch(e) { //exception
      print(e);
    }
  }) ,
```

Б қосымшасының жалғасы (міндетті)

```
    ],  
  ),  
),  
);  
}  
}
```

4 Басты бет

```
class NoteScreen extends StatefulWidget {  
  static String id = 'note_screen';  
  const NoteScreen({Key? key}) : super(key: key);  
  
  @override  
  State<NoteScreen> createState() => _NoteScreenState();  
}  
  
class _NoteScreenState extends State<NoteScreen> {  
  final _firestore = FirebaseFirestore.instance;  
  final _auth = FirebaseAuth.instance;  
  late User loggedInUser;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      backgroundColor: AppStyle.mainColor,  
      appBar: AppBar(  
        elevation: 0.0,  
        title: Text("HeyDay"),  
        centerTitle: true,  

```

Б қосымшасының жалғасы (міндетті)

```
backgroundColor: AppStyle.mainColor,
actions: <Widget>[
  IconButton(
    icon: Icon(Icons.close),
    onPressed: () {
      _auth.signOut();
      Navigator.pop(context);
    },
  ),
],
body: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        "Соңғы жазбаларыңыз",
        style: GoogleFonts.roboto(
          color: Colors.white,
          fontWeight: FontWeight.bold,
          fontSize: 22,
        ),
      ),
      SizedBox(
        height: 20.0,
      ),
    ],
  ),
),
```

Б қосымшасының жалғасы
(міндетті)

```
Expanded(
  child: StreamBuilder<QuerySnapshot>(
    stream:

FirebaseFirestore.instance.collection("Notes").snapshots(
),

    builder: (context,
AsyncSnapshot<QuerySnapshot> snapshot) {
      if (snapshot.connectionState ==
ConnectionState.waiting) {
        return Center(
          child: CircularProgressIndicator(),
        );
      }
      if (snapshot.hasData) {
        return GridView(
          gridDelegate:
SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2),
          children: snapshot.data!.docs
            .map((note) => noteCard(() {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context)
=>

NoteReaderScreen(note),

                ));
            });
      }
    },
  ),
);
```


Б қосымшасының жалғасы (міндетті)

```
        }, note))
        .toList(),
    );
}
return Text(
    "Жазбалар жоқ",
    style: GoogleFonts.nunito(color:
Colors.white),
);
},
),
),
IconButton(
    color: Colors.blue,
    onPressed: () {
        Navigator.push(context,
            MaterialPageRoute(builder: (context)
=> Calendar())));
    },
    icon: Icon(Icons.calendar_today_rounded),
),
IconButton(
    color: Colors.blue,
    onPressed: () {
        Navigator.push(context,
            MaterialPageRoute(builder: (context)
=> Pomodoro())));
```

Б қосымшасының жалғасы (міндетті)

```
    },  
    icon: Icon(Icons.timer),  
  ),  
  /*MaterialButton(  
    onPressed: () {  
      Navigator.push(context,  
        MaterialPageRoute(builder: (context)  
=> Calendar())));  
    },  
    minWidth: 200.0,  
    height: 42.0,  
    child: Text(  
      'Calendar',  
      style: TextStyle(  
        backgroundColor: Colors.blue,  
        color: Colors.white,  
      ),  
    ),  
  ), */  
],  
),  
),  
floatingActionButton: FloatingActionButton.extended(  
  onPressed: () {  
    Navigator.push(context,  
      MaterialPageRoute(builder: (context) =>  
NoteEditorScreen())));
```

Б қосымшасының жалғасы (міндетті)

```
    },  
    label: Text("Ескертпе қосу"),  
    icon: Icon(Icons.add),  
  ),  
);  
}  
}
```

5 Ескертпе қосу беті

```
class NoteEditorScreen extends StatefulWidget {  
  const NoteEditorScreen({Key? key}) : super(key: key);  
  
  @override  
  
  State<NoteEditorScreen> createState() =>  
    _NoteEditorScreenState();  
}  
  
class _NoteEditorScreenState extends  
State<NoteEditorScreen> {  
  int color_id =  
    Random().nextInt(AppStyle.cardsColor.length-1)+1;  
  
  String date = DateTime.now().toString();  
  
  TextEditingController _titleController =  
    TextEditingController();  
  
  TextEditingController _mainController =  
    TextEditingController();  
  
  @override
```

Б қосымшасының жалғасы (міндетті)

```
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: AppStyle.cardsColor[color_id],  
    appBar: AppBar(  
      backgroundColor: AppStyle.cardsColor[color_id],  
      elevation: 0.0,  
      iconTheme: IconThemeData(color: Colors.black),  
      title: Text(  
        "Жаңа ескертпе қосыңыз",  
        style: TextStyle(color: Colors.black),  
      ),  
    ),  
    body: Padding(  
      padding: const EdgeInsets.all(16.0),  
      child: Column(  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
          TextField(  
            controller: _titleController,  
            decoration: InputDecoration(  
              border: InputBorder.none,  
              hintText: 'Ескертпе тақырыбы',  
            ),  
            style: AppStyle.mainTitle,  
          ),  
          SizedBox(  
            height: 8.0,
```

Б қосымшасының жалғасы
(міндетті)

```
),
Text(
  date,
  style: AppStyle.dateTitle,
),
 SizedBox(
  height: 8.0,
),
 TextField(
  controller: _mainController,
  keyboardType: TextInputType.multiline,
  maxLines: null,
  decoration: InputDecoration(
    border: InputBorder.none,
    hintText: 'Жазба мазмұны',
  ),
  style: AppStyle.mainContent,
),
],
),
),
floatingActionButton: FloatingActionButton(
  backgroundColor: AppStyle.accentColor,
  onPressed: () async {
    FirebaseFirestore.instance.collection("Notes").add({
```

Б қосымшасының жалғасы
(міндетті)

```
"note_title": _titleController.text,  
"creation_date": date,  
"note_content": _mainController.text,  
"color_id": color_id,  
)).then((value) {  
  print(value.id);  
  Navigator.pop(context);  
}).catchError(  
  (error) => print("Failed to add new Note due  
to $error"));  
},  
child: Icon(Icons.save),  
),  
);  
}  
}
```