ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Тұрғынбек Бағынұр

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

6B06102 – Computer Science білім беру бағдарламасы

Алматы 2025

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
ПИ кафедрасының меңгерушісі
техн. ғыл. канд., қауымдастырылған
профессор
_____Ф.Н. Абдолдина
«____»_____2025ж.

ТҮСІНДІРМЕ ЖАЗБА
дипломдық жобаға

Тақырыбы: Learn to Easy – Android оқыту курстарына арналған қосымша

6B06102 – Computer Science білім беру бағдарламасы

Орындады Тұрғынбек Бағынұр

Рецензент
техн. ғыл. канд.
_____ Сейлова Н.А.
«_____»_____ 2025 ж.

Ғылыми жетекшісі
техн. ғыл. канд., қауымд. профессор
_____ Бейсембекова Р.Н.
«____»_____2025 ж.

Алматы 2025

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН
ПИ кафедрасының меңгерушісі
техн. ғыл. канд., қауымд. профессор
_____Ф.Н. Абдолдина
«____»_____2025 ж.

Дипломдық жобаны орындауға
ТАПСЫРМА

Білім алушыларға *Тұрғынбек Бағынұр*
Тақырыбы: Learn to Easy – Android оқыту курстарына арналған қосымша

Академиялық мәселелер жөніндегі проректор бұйрығының №_____«___» _____2025 ж. шешімімен бекітілген.
Орындалған жобаның өткізу мерзімі: *«___» _____ 2025 ж.*
Дипломдық жобаның бастапқы мәліметтері:
а) пәндік саланы талдау, ұқсас жобаларды талдау;
б) техникалық тапсырманы әзірлеу;
в) бағдарламаның архитектурасын әзірлеу;
г) түсінікті және интерактивті дизайнды жобалау және әзірлеу;
д) бағдарламалық жасақтаманы әзірлеу және тестілеу;
Дипломдық жобада әзірленуге жататын сұрақтардың тізбесі: (міндетті сызбаларды дәл көрсете отырып): *презентацияның ____ слайдтары ұсынылды.*
Ұсынылатын негізгі әдебиеттер: ___ *пайдаланылған әдебиеттер тізімінен.*

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	**Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі**	**Ескерту**
1. Дипломдық жоба тақырыбын қарастыру және әдебиеттер оқу	01.12.2024	Орындалды
2. Зерттеліп отырған тақырыптың жалпы сипаттамасын моделін құрау	05.12.2024	Орындалды
3. Әзірленген бағдарламалау ортасын таңдау	02.02.2025	Орындалды
4. Дипломдық жобаның мобильді қосымшасын дайындау және іске асыру	29.03.2025	Орындалды
5. Қосымша материалдар мен презентацияны дайындау	25.05.2025	Орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	**Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)**	**Қолтаңба қойылған мерзімі**
Бағдарламалық бөлім	Қабылжан Марғұлан техн.ғыл.магистр, аға оқытушы	
Нормалық бақылаушы	Нұралықызы Салтанат техн.ғыл.магистр, аға оқытушы	

Ғылыми жетекші _____ Бейсембекова Р.Н.

Тапсырманы орындауға қабылдап алған студент _____ Тұрғынбек Бағынұр

Күні «_ _» _____2025 ж.

АҢДАТПА

«Learn to Easy» мобильді қосымшасын әзірлеу білім беру мен ойын-сауық саласындағы отандық бағдарламалық өнімдерді дамытуда маңызды рөл атқарады. Flutter және Firebase технологияларын біріктіре отырып, жоба пайдаланушыларға топтық каналдар, жеке чаттар, хабарламалар және тесттер құрастыру мүмкіндігін ұсынатын бірегей платформа болып табылады. Қолданушылар өз тақырыптарындағы каналдарда пікір алмасып, посттар мен файлдар бөлісе алады, сондай-ақ он сұраққа дейінгі интерактивті квиздер жасап, нәтижелерін пайыздық көрсеткішпен бақылай алады.

Архитектурасында MVVM-тілдес қағидалар қолданылып, Flutter-дегі UI, бизнес-логика және деректер бөлек модульдерге бөлінген, бұл кодтың құрылымын қарапайым әрі кеңейтуге қолайлы етеді. Cloud Firestore-дың шынайы-уақытты синхрондау мүмкіндігі мен Impeller қозғалтқышының жоғары өнімділігі интерфейстің жылдам әрі мінсіз жұмыс істеуін қамтамасыз етеді. Пайдаланушының тіл мен тақырып таңдау (Light, Dark, Dracula) Shared Preferences арқылы құрылғы жадына сақталып, әрбір келесі іске қосу кезінде автоматты түрде қалпына келеді.

АННОТАЦИЯ

Разработка мобильного приложения «Learn to Easy» является значимым шагом в продвижении отечественного софта для образования и развлечений. Проект, основанный на Flutter и Firebase, представляет собой комплексную платформу, объединяющую групповые каналы, личные чаты, публикацию сообщений и конструктор тестов. Пользователи могут создавать дискуссионные каналы, обмениваться постами и файлами, а также генерировать интерактивные квизы до десяти вопросов с отображением результатов в процентах.

В архитектуре приложения воплощён подход, близкий к MVVM: пользовательский интерфейс, бизнес-логика и уровни данных разделены на независимые модули, что упрощает поддержку и расширение функционала. Cloud Firestore обеспечивает надёжное хранение и синхронизацию данных, а движок Impeller от Flutter гарантирует плавную и отзывчивую работу интерфейса. Параметры языка и темы оформления (Light, Dark, Dracula) сохраняются локально с помощью Shared Preferences и автоматически восстанавливаются при следующем запуске.

ANNOTATION

The development of the «Learn to Easy» mobile application marks a significant advancement in promoting home-grown software for education and entertainment in Kazakhstan. Built with Flutter and Firebase, the project offers a unified platform that integrates group channels, private chats, post sharing, and a quiz builder. Users can initiate thematic channels, collaborate by posting messages and files, and create interactive tests of up to ten multiple-choice questions with percentage-based scoring.

The application's architecture follows an MVVM-inspired separation of concerns, isolating the user interface, business logic, and data layers into modular components for easier maintenance and scalability. Cloud Firestore delivers secure, real-time data synchronization, while Flutter's Impeller rendering engine ensures a smooth and responsive UI. Language and theme preferences (Light, Dark, Dracula) are persisted locally using Shared Preferences and automatically restored on subsequent app launches.

МАЗМҰНЫ

КІРІСПЕ

Цифрлы дәуірдегі білім беру саласында мобильді қосымшалар ерекше рөл атқара бастады. Оқу процесін тиімді басқару, оқу материалын тез әрі ыңғайлы жеткізу, сондай-ақ оқушылар мен студенттердің мотивациясын арттыру мақсатында заманауи технологиялар белсенді түрде қолданылады. Қазақстандық білім беру кеңістігі де жыл өткен сайын цифрландыруға бет бұрып, дәстүрлі сынып практикасын онлайн режиммен үйлестіру қажеттілігін сезінуі тиіс. Дегенмен, қазіргі нарықтағы жетекші платформалар – Microsoft Teams, Zoom, Google Classroom – көбіне жаһандық стандарттарға негізделген, олардың интерфейсі мен функционалы қазақ тілін толықтай қолдамайды және жергілікті оқыту ерекшеліктерін ескеруі аз.

«Learn to Easy» жобасы осы мәселелерді шешуге бағытталған инновациялық мобильді платформа болып табылады. Ол Flutter фреймворкі мен Firebase қызметтерін пайдаланып, толықтай қазақ тілінде дайындалған интерфейс ұсынады[1]. Қосымша топтық каналдар, жеке чаттар, мәтіндік хабарламалар, интерактивті тестер мен статистиканы бақылау мүмкіндігін біріктіреді. Мұндай тұтас жүйе оқытушылар мен оқушылар арасында коммуникацияны жеңілдетіп, білім беру үрдісін жандандырады.

Архитектуралық тұрғыдан «Learn to Easy» платформасы модульдік құрылымды ұстанады: каналдар бөлімінде пайдаланушыларға өз пәндік топтарын құруға, материалдар мен хабарламалармен бөлісуге, талқылаулар ұйымдастыруға жағдай жасалады. Тесттер бөлімі оқытушыға максимум он сұрақтан тұратын викториналар дайындап, дұрыс жауапты бір ғана опция арқылы белгілеуге мүмкіндік береді, ал оқушылар өз білім деңгейін пайызбен бағалай алады. Жеке чаттар модулі қосымшаны әлеуметтік желідегі жеке хабар алмасу құралдарына ұқсас етіп, оқушылар мен оқытушыларға жеке мәселелерді талқылауға, сұрақ қоюға ыңғай береді.

Интерфейс дизайнында қолайлылық пен қарапайымдылық басымдыққа ие. Қолданушылар Light, Dark және Dracula тақырыптары арасында таңдау жасай алады, ал тіл параметрлері Shared Preferences арқылы құрылғыда сақталып, әр келесі іске қосуда автоматты түрде қалпына келеді. Cloud Firestore[7] дерекқорын қолдану шынайы-уақытты синхрондауды қамтамасыз етіп, интернет жылдамдығы төмен ортада да тұрақты жұмыс істеуге септігін тигізеді.

Осылайша, «Learn to Easy» жобасы Қазақстанның білім беру және edutainment нарығында өзекті шешім ретінде пайда болып, қазақ тіліндегі цифрлық оқыту платформалары арасында ерекше орын алады. Платформа оқыту үдерісін тиімді басқарып қана қоймай, оқушылардың өз білімін жүйелі сараптап, жеке нәтижелеріне талдау жасауға мүмкіндік береді. Бұл жоба оқыту мен ойын-сауық элементтерін үйлестіріп, қазақша білім беру экожүйесінің дамуына айрықша үлес қосады.

1 Өзектілігі

Қазіргі таңда онлайн оқыту платформалары білім беру процесінің ажырамас бөлігіне айналды. Дәстүрлі сыныптық сабақ үлгісін толық алмастыруы мүмкін болмаса да, пандемиялық кезең мен цифрландыру үрдістері оқу үдерісін онлайн формаға тез арада бейімдеуге мәжбүр етті. Оқытушылар үшін платформалар интерактивті тапсырмалар құрастыру, сабаққа қатысушылар санын бақылау, файлдар мен презентациялармен бөлісу мүмкіндігін ұсынады. Оқушылар мен студенттерге қарапайым чат, бейнеқоңырау, тесттер және тапсырмалар тізбегін орындау арқылы оқу материалын меңгеруді жеңілдетуге көмектеседі.

Нарықтағы жетекші шешімдер – Microsoft Teams, Zoom, Google Classroom – халықаралық стандарттарға сәйкес жоғары сенімділік пен функционалдық мүмкіндіктер ұсынып отыр. Дегенмен, олар көбіне жалпыға ортақ талаптар мен түсініктерге негізделіп жасалғандықтан, қазақ тілі мен Қазақстанның білім беру ерекшеліктерін толықтай ескермейді. Мысалы, Microsoft Teams интерфейсінің түбірінен күрделі құрылымы, кейбір аудандағы интернет жылдамдығына ықпал ете алмайды; Zoom платформасының бейнеқоңырау сапасы жылдам желіден айырылғанда нашарлайды; ал Google Classroom–дағы тест құрастыру мүмкіндіктері кейде шектеулі болып, тек Google-дің экожүйесіне тәуелді.

Сонымен қатар, мұндай жаһандық шешімдер локализация деңгейінде де кемшіліктерге ұшырайды. Қазақ тіліндегі толық аудармасы болмағандықтан, жергілікті оқытушылар мен оқушылар интерфейсті түсінуде қиындыққа тап болуы мүмкін. Оған қоса, Қазақстанның ұлттық контенті мен оқыту әдістемелерін біріктіру мүмкіндігі шектеулі: мемлекеттік білім стандарттары, тест сұрақтарының типтері және ұлттық емтихан форматтары (мысалы, ҰБТ) арнайы бейімделуді талап етеді.

Осы орайда, «Learn to Easy» жобасының өзектілігі айқын көрінеді. Қазақ тілінде жасалған, толық локализацияланған, мемлекеттік және мектеп деңгейіндегі оқу-әдістемелік талаптарға сәйкес үйлестірілген платформа ретінде ол отандық білім беру экожүйесін толықтырады. Платформа келесі маңызды артықшылықтарға ие болады:

1. **Мемлекеттік білім стандарттарына сәйкестік.** Құрылымында Қазақстан Республикасының жалпы орта білім беру мен жоғары оқу орындарының оқу жоспарындағы пәндер мен модульдерге сәйкес тесттер мен тапсырмалар жүйесін орналастыруға мүмкіндік бар. Мысалы, ҰБТ форматындағы сұрақтар жинағын жасау немесе мектеп курстық бағдарламасын қолдау.
2. **Толықтай қазақ тіліндегі интерфейс.** Пайдаланушыға таныс, ана тілімізде жасалған мәтіндік және бейне материалдар, аудармалардың сапасы жоғары болғандықтан қолжетімділік артады.
3. **Қолжетімділік және ресурстарды оңтайлы пайдалану.** Flutter негізінде әзірленген мобильдік қосымша Android және iOS платформаларында

ұқсас өнімділікпен жұмыс істейді. Импеллер (Impeller) рендеринг моторы және Firebase-дың шынайы уақытта синхрондау мүмкіндігі жадыны аз пайдаланып, интернет жылдамдығы төмен ортада да тұрақты қондырма ұсынады.

4. **Интерактивті оқыту құралдары.** Тапсырмалар мен тесттерді құрастыру кезінде оқытушыларға 10 сұраққа дейінгі квиздер жасауға, әрбір сұраққа төрт нұсқа ұсынуға, бір ғана дұрыс жауапты таңдау мүмкіндігін беріп, нәтижелердің пайыздық көрсеткішін автоматты түрде есептеп шығарады. Мұндай құрылым оқушылардың білім деңгейін объективті бағалауды жеңілдетеді.

5. **Жеке және топтық чаттар.** Оқушылар оқытушылармен немесе бір-бірімен оқыту барысы, сұрақтар мен қиындықтар бойынша жеке сөйлесе алады. Бұл оқу үдерісін жедел әрі жекелендірілген түрде жүргізуге мүмкіндік береді.

6. **Жергілікті қоғамдық трендтер мен мәдени ерекшеліктерді ескеру.** Қосымшаға Қазақстанның мектеп мәдениеті мен оқу процесіне тән мерекелер, іс-шаралар, ұлттық контенттерді енгізу арқылы оқушылардың танымын кеңейтуге бағытталған.

Осылайша, «Learn to Easy» платформасы – халықаралық сапа стандарттарын сақтай отырып, Қазақстандық білім беру нарығының қажеттіліктеріне бейімделген, толықтай қазақ тіліне локализацияланған шешім. Бұл жоба еліміздегі онлайн оқыту платформаларының санын арттырып қана қоймай, сапаны және оқыту әдістемесін жаңа деңгейге көтеруге, қазақ тілінің білім беру саласында кеңінен пайдаланылуына ықпал етеді.

1.1 Жобаның мақсаты

«Learn to Easy» мобильді қосымшасының басты мақсаты – қашықтан оқыту мен коммуникация саласында Қазақстандық білім беру экожүйесіне арнайы бейімделген, қарапайым әрі қолжетімді шешім ұсыну.

Дипломдық жоба шеңберінде мынадай негізгі бағыттар анықталды:

1. Қашықтан оқыту мүмкіндігін кеңейту және үнемі қолжетімді ету. Қазіргі заманғы білім беру үрдісінде мектептер мен жоғары оқу орындары дәстүрлі аудиториялық сабақтарды сырттай оқыту форматымен ұштастыруда. Алайда, Ұлытау өңірі сияқты шалғай аймақтардағы интернет желісінің тұрақсыздығы немесе оқытушы мен студенттің жиі сапарға шығуы оқыту процесіне кедергі келтіреді. «Learn to Easy» бұл мәселені мобильді құрылғылар арқылы шешеді: сабақтар, тапсырмалар және чаттар смартфон, планшет немесе басқа да портативті құрылғыларда тұрақты түрде қолжетімді болып, пайдаланушыларға кез келген уақытта білім алуға мүмкіндік береді.

2. Қазақша отандық платформа әзірлеу.

«Learn to Easy» жобасы – толықтай қазақ разработчиктер тобы әзірлеген, Қазақстан нарығына арналған отандық шешім. Бүгінгі таңда көбіне шетелдік IT-компаниялардың қолдауымен жасалған платформалар қолданылады: олардың

интерфейсі мен архитектурасы глобалдық аудиторияға бейімделсе де, отандық бағдарламашылардың мүддесі мен локальді талаптарды есепке алмайды. «Learn to Easy» – керісінше, Қазақстандық студенттер мен оқытушылар үшін нақты қажеттіліктерге сай жасалған программалық өнім.

Барлық мәзір элементтері, ескертпелер мен көмекші мәтіндер бастапқыдан-ақ қазақ тілінде жазылып, кәсіби лингвист мамандармен тексерілген. Бұл қолданушыларға жүйені тез үйренуге және күнделікті оқыту үдерісінде шатасып қалмауға мүмкіндік береді. Сонымен қатар, қосымшаның кодтық базасы мен UI/UX дизайны жергілікті әзірлеушілердің тәжірибесіне негізделіп, мобильдік және веб-қосымшалар арасында бірізділік сақтайды.

Пайдаланушылар интерфейс элементтерін қазақ тілінде көріп қана қоймай, платформаның толықтығымен де мақтана алады: барлық функционалдық модульдер (чат, каналдар, тапсырмалар, тесттер) Қазақстанның білім беру стандарттары мен техникалық инфрақұрылым ерекшеліктеріне сәйкес оңтайландырылған. Бұл жоба – қазақ софтын дамытуда жаңа бағыт ашып, білім беру саланың цифрландыруына өз үлесін қосатын инновациялық бастама.

3. Оқытушылар мен студенттер арасындағы өзара әрекеттестікті тиімділету.

Жобада оқу құралы ретінде тек статикалық тесттер мен мәтіндік хабарлар ғана емес, сонымен қатар интерактивті тапсырмалар мен чат жүйесі қарастырылған. Оқытушылар тапсырма жіберу, кері байланыс беру, жеке немесе топтық талқылау ұйымдастыру мүмкіндігін алады. Студенттер болса, сұрақ қою, жауап беру, тест нәтижелерін талдау арқылы өзіндік білім алу жолын қадағалап отырады. Болашақта мұнда бейнеконференциялар, құжаттармен бірлескен жұмыс және білім беру аналитикасын енгізу жоспарланып отыр, бұл жобаға кеңейтілген функционал қосудың әлеуетін көрсетеді.

4. Халықаралық әдістемелерді ескере отырып, ұлттық оқу стандарттарын біріктіру.

Қазақстан Республикасының орта білім беру және жоғары оқу орындары бағдарламалары Ұлттық ақпараттық жүйеде тіркелген және мемлекеттік стандарттармен бекітілген. «Learn to Easy» платформасына ҰБТ үлгісіндегі сұрақтар мен жауаптар, пәндерге арналған бейімделген квиздер енгізіледі. Бұл әдістемелік негізді пайдалану Қазақстандағы білім беру үрдісін цифрлы форматқа ауыстыруда практикалық маңызға ие.

5. Білім беру нарығында бәсекеге қабілетті отандық өнім ұсыну.

Әлемдік edtech нарығындағы көшбасшы платформалар жылдар бойы дамып, ауқымды функционал жинағанда, отандық шешімдер әлі толыққанды бәсекелес бола алмай келеді. «Learn to Easy» мобильді қосымшасы қарапайымдылығы мен қазақ тіліне бейімделу ерекшелігі арқасында ел ішінде кеңінен таралып, кейіннен көрші елдерде де қолданылуға мүмкіндігі бар. Жоба Қазақстандағы стартап экожүйесіне де серпін беріп, технологиялық салада жаңа жұмыс орындарын ашуға ықпал ете алады.

6. Пайдаланушы тәжірибесін үздіксіз жетілдіру және озық технологияларды интеграциялау.

Жоба алғашқы кезеңде негізгі функцияларға (каналдар, посттар, тесттер, чаттар) назар аударса, келесі кезеңдерде жасанды интеллект негізінде бейімделген оқу ұсыныстарын, дауыстық танып білу және мәтінді аудару жүйелерін, виртуалды зертханалар мен интерактивті тақталарды енгізу жоспарланып отыр. Осы мүмкіндіктерді қосу білім алушының жеке ерекшеліктеріне қарай мазмұнды бейімдеуге мүмкіндік береді және жобаның дамуы үшін айрықша әлеует қалыптастырады.

7. Оқу үрдісін анализдеу және статистикалық есептер беру.

Cloud Firestore және Firebase Analytics қолдану арқылы пайдаланушы әрекеттерін жинап, сабаққа қатысу деректері, тест нәтижелері, чаттағы белсенділік статистикасын автоматты түрде есептеу жүйесіне енгізуге болады. Болашақта оқытушылар мен әкімшілік топқа оқушы үлгерімі туралы кешенді талдау құралдарын ұсыну жобаның білім беру процесін басқарудағы практикалық құндылығын арттырады.

8. Қауіпсіздік пен деректерді қорғау.

Жоба шеңберінде пайдаланушының жеке деректері мен оқу нәтижелері сенімді түрде сақталуы үшін Firebase Authentication, Firestore Security Rules және шифрлау технологиялары қолданылады. Пайдаланушылардың өз профилі мен чат тарихын қорғау жоғары деңгейдегі қауіпсіздік талаптарына сәйкес орындалады.

9. Пайдалануға ыңғайлылығы мен масштабталуы.

Flutter технологиясы қосымшаны Android және iOS жүйелерінде бірдей өнімділікпен іске қосуға мүмкіндік береді. Код модулі архитектурасы жаңа функциялар мен интернационализацияны оңай енгізуге жағдай жасайды, әрі болашақтағы жаңартуларды жылдам орындауға мүмкіндік туғызады.

Осылайша, «Learn to Easy» жобасының мақсаты – Қазақстандық білім беру нарығында жоғары бәсекеге қабілетті, толықтай қазақ тілінде локализацияланған, заманауи технологиялар мен халықаралық әдістемені үйлестірген мобильді оқыту платформасын жасау. Бұл жоба оқыту үрдісін тиімді басқаруға, оқытушылар мен студенттер арасындағы өзара әрекеттестікті арттыруға, сондай-ақ цифрлық білім беру саласындағы отандық шешімдерді дамытуға бағытталған кешенді әрі перспективалы өнімге айналуға ұмтылады.

1.2 Тапсырмалар

«Learn to Easy» жобасында негізгі функционалдық міндеттер толықтай анықталып, MVP (Minimum Viable Product) деңгейіндегі прототипті жүзеге асыру үшін келесі тапсырмалар қарастырылды. Әрбір тармақ бойынша қазіргі қолданыстағы шешімдер сипатталып, болашақта кеңейту мен жетілдіру мүмкіндіктері талданады.

Интуитивті интерфейс әзірлеу.

Қолданушы ағынын жеңілдету үшін «RegisterStep1Screen» және «LoginScreen» -де қарапайым форма валидациясы қосылды: тіркелу мен кіру өрістері тексеріліп, қате енгізулерге жадыратулар көрсетіледі.

«BottomNavBar» виджеті арқылы негізгі экрандар арасына жылдам өту мүмкіндігі ұсынылған. Экрандардың атаулары және иконкалары қолданушыға қай бөлімде тұрғанын дәл көрсетеді.

«PersonScreen» (жеке профиль) және «SettingsScreen» бөлімдерінде қолданушы өз деректері мен параметрлерін бір жерден өзгертіп, Live Preview режимінде тақырып пен тіл өзгерістерін көре алады.

Дамушы әлеует: drag-&-drop элементтерін қосып, басты мәзірді және тапсырмалар тізімін жеке бейімдеуге мүмкіндік беру; swipe-жесттер арқылы экрандар арасында жылдам ауысу; onboarding (қосымшаны алғаш ашқанда) интерактивті нұсқаулық енгізу.

Онлайн және офлайн форматта сабақ өткізу мүмкіндігін қамтамасыз ету.

Қазіргі таңда «ChannelScreen» экранында Cloud Firestore-дан оқиғалар (posts, tests, messages) realtime режимінде жазылып, оқылып отырады. Бұл онлайн сабақтарды ұйымдастыруға жарамды.

Офлайн режим: Firestore SDK-ның offline persistence мүмкіндігі қосылғанымен, UI деңгейінде офлайн сақталған деректерді көрсету, синхрондау кезегі және қақтығыстарды шешу логикасы әлі толық жазылған жоқ.

Дамушы әлеует:

Offline Queue — желі жоқ кезде жіберілген хабарламалар мен посттарды локалдық дерекқорға (SQLite, Hive) сақтап, интернет пайда болған соң автоматты түрде серверге синхрондау.

Кэш саясаттары — соңғы 50 чат хабарламасын, 10 соңғы тест нәтижесін локалдық жадыда сақтау.

Sync Status индикаторы — деректер синхрондалған/синхрондалмағанын қолданушыға көрсету.

Тапсырмалар мен бағалау жүйесін енгізу

«Tasks» табында тест құрастыру модулі («Create Test» түймесі) және «TestDetailScreen» арқылы пайдаланушылардың жауаптары мен пайыздық көрсеткіштері сақталады. Әр тест нәтижесі «results» подколлекциясына жазылады.

Бағалау жүйесі: қазіргі нұсқада әр сұрақ тек бір дұрыс жауаппен бағаланып, жалпы пайыз санау жүзеге асырылған.

Дамушы әлеует:

Асинхронды тапсырмалар (homework) – оқытушы белгілі бір тестті немесе эссені белгілеп, аяқтау мерзімін тағайындауы.

Кері байланыс – тест жауаптары автоматты бағаланғаннан кейін оқытушы ескерту мәтіні мен баға қосу мүмкіндігі.

Жылдық/айлық статистика – әр сабақ, тақырып бойынша орташа балл, белсенділік индикаторы.

Firebase негізінде деректерді қауіпсіз сақтау және синхрондау

Firebase Authentication[7] арқылы тіркелу/кіру, Cloud Firestore-да channels, posts, tests, results, privateChats, messages коллекциялары қолданылған.

Firestore Security Rules конфигурацияланып, оқу-жазу құқығы қатысушылар мен көзі таза авторларға берілген.

Дамушы әлеует:

Cloud Functions – триггерлік функциялар жазу (тест нәтижесі шыққанда оқытушыға хабарландыру).

Firebase Analytics – қолданушы әрекеттерін талдау, белсенділік траекториясын анықтау.

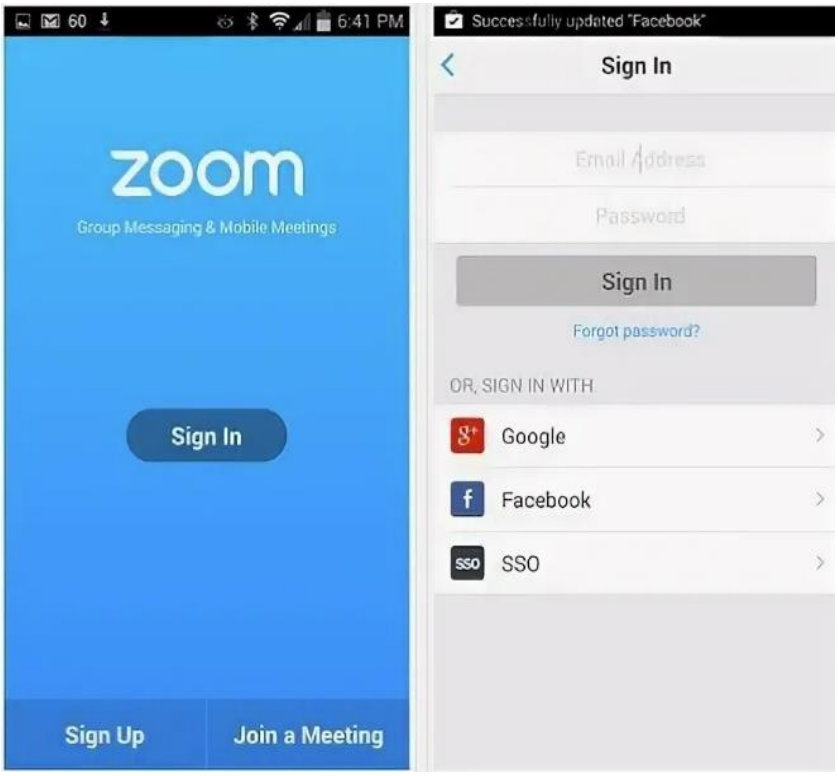
Crashlytics – апаттар мен қатені мониторингтеу, сапаны арттыру.

1.3 Ұқсас жобаларға талдау жасау

Microsoft Teams – білім беру мен іскерлік коммуникация үшін әмбебап платформа. Оның артықшылығы – Office 365, SharePoint, OneDrive және басқа да Microsoft қызметтерімен терең интеграцияланғандығында. Мұнда каналдар, чаттар, құжаттарды бірге өңдеу, кездесулерді жоспарлау мүмкіндіктері бар. Алайда, Teams интерфейсі күрделі, жаңа қолданушыларға үйрену қиын, және лицензия талаптары университетке немесе шағын сыныпқа бейімделмейді. Сонымен қатар, тесттер мен тапсырмаларды автоматты түрде құру және бағалау функциялары жеткіліксіз, оқытушыларға әртүрлі сыртқы құралдарға жүгінуге тура келеді (1 сурет).

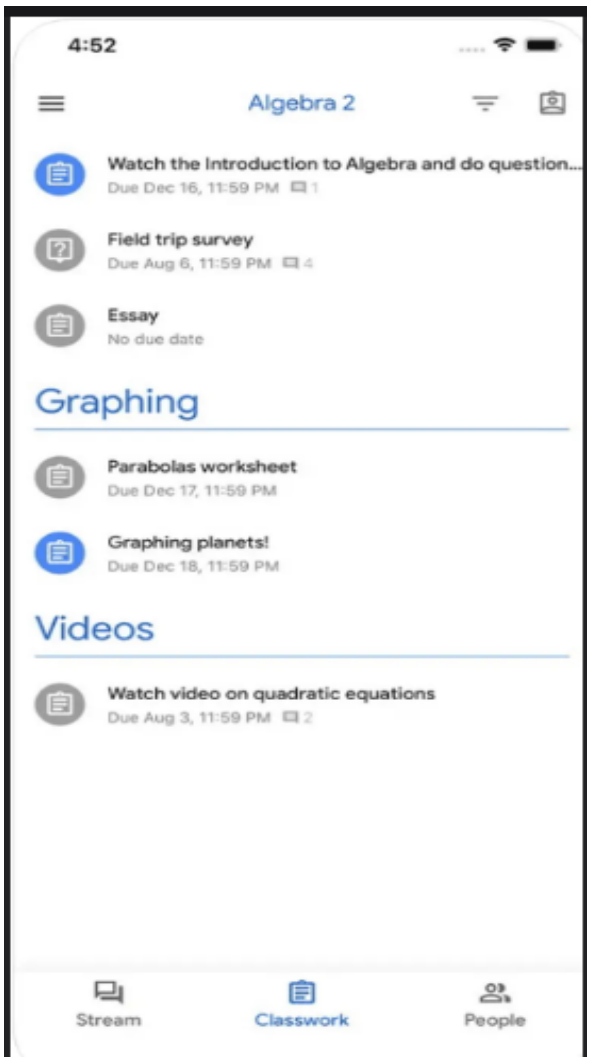
Zoom платформасы видеоконференциялар ұйымдастыруға ыңғайлы: экранды бөлісу, виртуалды фонын ауыстыру, «кіші бөлмелер» (breakout rooms) сияқты функциялар сабақ өткізу кезінде тиімді. Платформа сенімді, көп қолданушыны бір уақытта қолдайды. Алайда, Zoom-да оқу материалдарын жүйелі түрде басқару, тесттер, тапсырмалар мен бағалау жүйесі жоқ. Сабақ жазбасын бөлісу мүмкін болса да, Learning Management System (LMS) ретінде толыққанды жұмыс істемейді. Оқу процесін бір орталықтан бақылау үшін қосымша модульдер қажет.

1 сурет – Teams UI



2 сурет – Zoom UI

Google Classroom – мектептер мен колледждерге арналған қарапайым және жеңіл интерфейсімен танымал. Ол Google Drive, Docs, Sheets, Calendar интеграциясын пайдаланып, тапсырмаларды бөлісу мен орындауды жеңілдетеді. Оқытушылар материалдарды жүктеп, баға қоя алады, ал оқушылар тапсырманы электронды түрде тапсыра алады. Алайда, кіріктірілген чат функциясы жоқ, қатысушылардың өзара қарым-қатынасы комментарийге шектеледі. Видеоқоңырау үшін Google Meet-ке көшу керек, сондықтан бірқатар оқытушылар мен оқушылар үшін қосымша қиындықтар туындайды. Қазақша локализация толық емес, аудармаларда дәлдік аз.



3 сурет – Google Classroom Ui

Осы үш платформа білім беру нарығында мықты позицияға ие болғанымен, олардың барлығынан ортақ бір кемшілік табылады – олар көптілді қазақ аудиториясының ерекшеліктерін ескермейді. Қазіргі уақытта қазақша интерфейс пен терминологияның сапасы төмен, оқу-тәрбиелік контентті толықтай қазақша ұсынатын құралдар аз. Сонымен бірге, офлайн режимде

жұмыс істеу немесе төменгі жылдамдықтағы желіде тапсырма жіберу мүмкіндігі жоқтың қасы.

«Learn to Easy» қосымшасы осы олқылықтарды жоюға бағытталған. Біріншіден, толық қазақ тілінде жасалып, барлық мәзір элементтері, ескертулер мен көмекші мәтіндер ана тілінде ұсынылады. Екіншіден, платформада каналдар (топтық чат), жеке чат, тесттер мен тапсырмаларды құру, автоматты бағалау және статистика мүмкіндігі қарастырылған. Үшіншіден, Firebase-тің офлайн кеші мен синхрондау мүмкіндіктері арқасында интернеті жоқ жағдайда да бұрын жүктелген материалдарға қолжетімділік сақталады. Төртіншіден, бейне- және тексттік хабарламаларды бір жүйеде ұсына отырып, оқыту процесін жан-жақты қолдау көзделген.

Осылайша, Microsoft Teams-тің кеңейтілген корпоративтік құралдары мен Zoom-дың видеоконференция қарапайымдылығын, Google Classroom-ның тапсырма беру жүйесін біріктіре отырып, «Learn to Easy» нақты қазақ оқушылары мен оқытушылардың қажеттілігін толық қанағаттандыратын шешім ұсына алады. Бұл платформа жас ұрпақты білім алуға ынталандырумен қатар, жергілікті контент пен педагогикалық тәжірибені цифрлы ортаға тиімді енгізуге мүмкіндік береді.

1.4 Learn to Easy – айырмашылықтары

«Learn to Easy» платформасы нарықтағы танымал онлайн-оқыту құралдарынан бірнеше маңызды ерекшеліктерімен ерекшеленеді. Біріншіден, онда топтық және жеке қарым-қатынас Telegram стиліндегі ыңғайлы чат арқылы ұйымдастырылған. Оқытушылар мен студенттер бір арнаға (каналға) қосылып, мәтіндік хабарламалар, стикерлер мен файлдар бөлісе алады. Сонымен қатар, әрбір пайдаланушы басқа қатысушымен жеке сөйлесе алады: жеке чат ішінде оқытушы тапсырмаларды нақты бір оқушыға жібере алады не студент кері сұрақтарын жібере алады. Бұл формат оқытушыларға фокусталған кері байланыс беруге, ал студенттерге кез келген жерде және кез келген уақытта сұрақ қоюға мүмкіндік береді.

Екіншіден, «Learn to Easy» офлайн режимді толықтай қолдайды. Қосымша Flutter-тың және Firebase-тің офлайн кеші механизмдерін пайдалана отырып, сабақ материалдары мен тесттерді алдын ала жүктеуге және интернетсіз де оларға қол жеткізуге мүмкіндік береді. Бұл әсіресе ауылдық аймақтар мен тұрақты байланыс жоқ жерлерде тұратын оқушылар үшін аса маңызды. Оқу барысында жасалған жазбалар, тест жауаптары локалды құрылғыда сақталып, желі қосылған кезде автоматты түрде сервермен синхрондалады.

Үшіншіден, платформа Flutter және Firebase негізінде әзірленгендіктен, жылдам жүктелу, минималды кідіріс және сенімді деректер қоры қамтамасыз етіледі. Пайдаланушы интерфейсі iOS пен Android жүйелерінде бірдей

өнімділікпен жұмыс істейді, ал Firebase-тегі аутентификация мен қауіпсіздік ережелері деректердің тұтастығын сақтайды.[2]

Төртіншіден, «Learn to Easy» оқу нәтижелерін бақылау және талдау жүйесін қамтиды. Әрбір тапсырма мен тестің нәтижесі статистикаға тіркеліп, ооқытушыны оқушының ұпайы, прогресі және әлсіз бағыттары туралы нақты мәліметпен қамтамасыз етеді. Бұл оқыту процесін жекелей бейімдеуге және әр студенттің оқу траекториясын оңтайландыруға септігін тигізеді.

Осы аталған функциялар «Learn to Easy» жобасын оқыту платформалары арасында бірегей етеді: ол бір уақытта әлеуметтік байланыс, офлайн режим, жоғары өнімділік және прогресті талдау мүмкіндігін біріктірген кешенді шешім ұсынады.

2 Технологияны таңдау

2.1 Flutter Dart

Flutter – Google әзірлеген ашық бастапқы кодты кроссплатформалық фреймворк, ол бір ғана код базасымен Android, iOS, веб және десктопта жұмыс істейтін қосымшалар жасауға мүмкіндік береді. Flutter-дың негізінде Dart тілі жатыр: ол JIT (Just-In-Time) және AOT (Ahead-Of-Time) компиляцияны қолдайды, сондықтан әзірлеу кезінде «Hot Reload» арқылы өзгерістерді бірден көруге болады, ал шығарылым нұсқада қосымшаның жылдамдығы жоғары сақталады.[1]

Неліктен Flutter + Dart?

Кроссплатформалықтық. Flutter-дың басты артықшылығы – бір кодпен бірнеше платформаға шығарылым дайындау. Менің «Learn to Easy» жобасында біз Android және iOS қосымшаларын бірдей логикамен, бірдей UI-мен ұсынғымыз келеді. Мысалы, main.dart файлын қарасаңыз, онда қосымшаға кіріс нүктесі жазылған және Flutter-дың MaterialApp виджетін пайдалану арқылы бірден екі платформада да ортақ тақырыптар мен маршруттар анықталған.[3]

UI икемділігі. Flutter-да бәрі – виджет. Тіпті экран, батырма, мәтін – барлығы виджет. Мысалы, channel_screen.dart ішінде біз TabBar мен TabBarView пайдаланып, үш түрлі қойынды (Posts, Tasks, Files) жасай аламыз. Виджеттердің ағашы (widget tree) арқылы интерфейсімізді логикалық-бөлімшелерге бөлу жеңіл.[3]

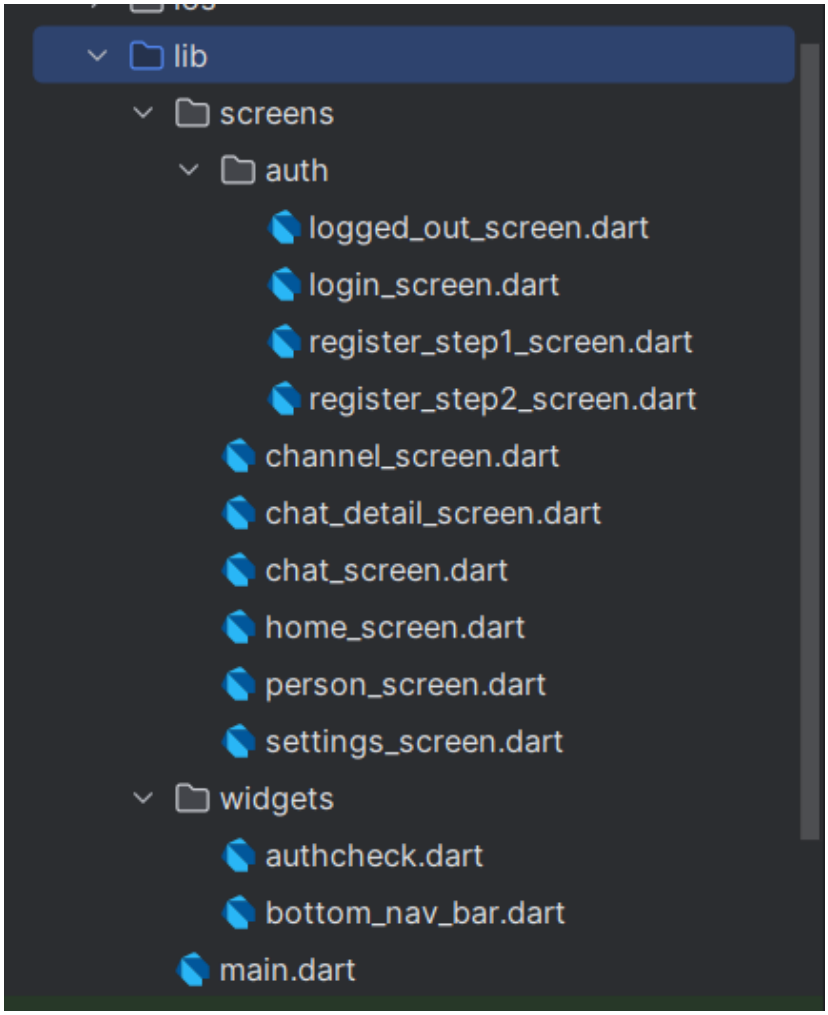
Жоғары өнімділік. Flutter өзінің Skia графикалық қозғалтқышын қолданады, ол GPU-ға тікелей сызып, анимациялар мен күрделі интерфейс элементтерін үзіліссіз көрсетеді. Dart AOT компиляциясымен компиляцияланған байт-код жылдам жүктеліп, Native жылдамдыққа жақындап, қолданбаның жауап беру уақытын азайтады.

Hot Reload. Flutter-дың «горячая перезагрузка» мүмкіндігі өзгертулерді секунд ішінде көруге мүмкіндік береді. Мысалы, мен bottom_nav_bar.dart виджетіне кейбір түстерді өзгертіп, жылдам тексердім – қолданба автоматты түрде қайта жүктеліп, стиль жаңарды; бұл уақыт үнемдейді және кодты интуитивті түсіруге көмектеседі.

Dart тілі және оның артықшылықтары.

Dart – міндетті типтеу және асинхронды programming model (async/await) ұсынатын тіл. Менің жобада Firebase-пен жұмыс істеу үшін асинхронды Future және Stream ұғымдары өте ыңғайлы болды: мысалы, authcheck.dart ішінде StreamBuilder<User?> қолдану арқылы қолданушы авторизация күйін бақылауға болады. Dart-тағы async/await синтаксисі кодты оқуға жеңіл, ал Promise-қа ұқсас Future арқылы желіден деректерді алу қарапайым әрі анық.

Project құрылымы Flutter-да



4 сурет – Жобаның құрылымы

main.dart – қосымшаның кіріс нүктесі, тақырып, навигация маршруттары тіркеледі.

widgets/ – қайта қолданылатын компоненттер: төменгі навигация жолағы (bottom_nav_bar.dart), авторизация күйін тексеретін AuthCheck (authcheck.dart).

screens/ – әрбір экран бөлек файлда орналасқан: логин, тіркелу, каналдар тізімі, чат, профиль, баптаулар. Мұндай логикалық бөлу кодты ұқыпты әрі масштабталатын қылады.

Flutter экожүйесі және пакеттер.

Жоба ішінде біз provider пакетін қолдандық (кейбір көрінбей қалған жерлерде), бірақ басты құрал – firebase_auth, cloud_firestore және firebase_core. Барлық Firebase пакеттерінің соңғы BoM-нұсқасы pubspec.yaml ішінде көрсетілген, бұл тәуелділіктерді басқаруды жеңілдетеді. Flutter CLI арқылы flutter pub get және IDE-лерде плагиндердің көмегімен бәрі оңай орнатылады.

Не дағдарыс, не қиындық болды?

Шын айтсам, Dart-та кейде типтермен күресу керек, әсіресе кеңейтілген StreamBuilder<QuerySnapshot> сияқты, онда .data түрлендіру мәселесі пайда болады. Мысалы, channel_screen.dart ішінде алдымен тек Object? алу қатесін жоятын as Map<String, dynamic> түрлендірулер қолданылды. Бірақ бұл тәжірибе – Dart-тың статикалық типтеуінің артықшылығы: қателерді ертерек табуға мүмкіндік береді. Flutter + Dart тәсілі «Learn to Easy» жобасын жылдам және сенімді жүзеге асыруға үлкен ықпал етті. Бір код базасы, бай виджет-көптігі, Hot Reload және жоғары өнімділік – осының бәрі дипломдық жобаны уақытында дайындауға және болашақта жаңа мүмкіндіктерді (тесттер, бейнеконференциялар, статистика визуализациясы) оңай қосуға көмектеседі. Менің ойымша, Flutter-дың экожүйесі мен Dart-тың мүмкіндіктері қазақша білім беру қосымшаларын дамытуда үлкен перспективалар ашады.[4]

2.2 Android Studio

Android Studio – Google әзірлеген ресми интеграцияланған даму ортасы, Android қосымшаларын құруға арналған толық IDE.[5] Flutter және Dart жобаларымен жұмыс істеу үшін арнайы плагиндер бар, сондықтан біз «Learn to Easy» жобасын дәл осы ортада құруды таңдадық.

IDE ішінде **Flutter** қолдауы белсенді: жоба жасау шеберінен бастап, құрылым құруға дейінгі барлық қадам Dart кодында автоматты орындалады. **Hot Reload** батырмасын басқаннан кейін жасалған өзгерістер секундына көрінеді, сондықтан UI-ді мұқият тексеру және түзету аса ыңғайлы.[3]

Интерактивті UI редакторы мен **Flutter Inspector** құралдары виджет ағашын визуалды түрде қарауға және элементтердің пропертиілерін динамикалы өзгертуге мүмкіндік береді. Мысалы, home_screen.dart ішіндегі ListView пен AlertDialog виджеттерін IDE арқылы жылдам реттеп, нәтижесін дереу көрдік.

Код жазуға арналған мүмкіндіктер де жоғары: **интеллектуалды автотолықтыру**, **синтаксис қателерін нақты көрсету**, **рефакторинг құралдары** арқасында кодтың сапасы артады. **Logcat**, **Memory Profiler**, **CPU Profiler** сияқты панельдер қосымшаның өнімділігін бақылауға көмектеседі.

Эмуляторлар мен **виртуалды құрылғылар** әртүрлі Android нұсқаларын сынаққа мүмкіндік береді. Біз Android 8.0, 11.0 үлгілерін қосып, әрбір экранның дұрыс көрінуін тексердік. Сондай-ақ, IDE ішіндегі **интеграцияланған терминал** және **Git** құралдары арқылы версионирлеу пен автоматты жинау (Gradle) үздіксіз жұмыс істеді.

Кемшілігі: Android Studio кейде ауырлап, көп жадты қажет етеді. Бірақ жоба көлемі артқан сайын бір ортада бәрін бақылап отыру үлкен артықшылық. Осылайша, Android Studio «Learn to Easy» проетінде сапасы мен тез дамуын қамтамасыз ететін негізгі құрал болды.

2.3 Firebase

Firebase – Google әзірлеген бұлтты платформа, ол мобильді және веб-қосымшаларды әзірлеушілерге сервер жағындағы инфрақұрылымды басқармай-ақ толыққанды функционал ұсынуға мүмкіндік береді.[7] Батыста “backend as a service” (BaaS) деп аталады, яғни “серверсіз” шешім саналады.[11]

Firebase негізгі компоненттері

Cloud Firestore & Realtime Database

Cloud Firestore – құжат-жұптық (document-based) құрылымды NoSQL дерекқор, онда деректер «коллекциялар» мен «документтер» түрінде сақталады. Firestore нақты уақыттағы (real-time) синхрондау үшін Stream API ұсынады: Flutter-дегі StreamBuilder<QuerySnapshot> арқылы қосымша автоматты түрде жаңартылады.[11]

Realtime Database – ескі нұсқадағы JSON-бейімді дерекқор, бірақ тез жауап беруі керек жағдайларда қолданылады. Біз «Learn to Easy» жобасында жөндеу жұмыстары барысында негізінен Firestore қолдандық, ал Realtime Database-ті чат тарихын кеш синхрондау немесе жеңіл құрылымды хабарламалар үшін қосымша енгізуге болады.

Authentication (Аутентификация)

Firebase Authentication Google, Facebook, Apple, Email/Password, телефон нөмірі секілді танымал провайдерлерді қолдайды. Қатты конфигурация жазбай-ақ, firebase_auth плагині арқылы Flutter-де тек signInWithEmailAndPassword немесе signInWithGoogle шақырсаң болды – қолданушы дайын тіркеледі.

Пәрмен жолында немесе консольде аутентификация әдістерін қосып, тексеріп отыруға ыңғайлы.

Cloud Messaging (Push хабарламалар)

Firebase Cloud Messaging (FCM) арқылы Android және iOS құрылғыларына хабарлама жіберуге болады. Оқытушылар тест жариялаған кезде, жаңа канал қосылғанда немесе жеке хабар келгенде пуш-хабарлама алып отыратын боламыз.

Cервepless функциялар (Cloud Functions) негізінде триггер жазып, құжат қосылғанда автоматты түрде FCM шақыруға болады.

Storage & Hosting

Firebase Storage – үлкен файлдарды (бейне, суреттер, PDF) сақтау үшін қолданылады. Пайдаланушы профиль суретін немесе сабақ материалдарының PDF құжаттарын жүктеу қажет болса, сол жерде сақтаймыз.

Firebase Hosting – статикалық веб мазмұнын (жоба сайты, документация) орналастыруға арналған.[8]

Analytics, Crashlytics, Performance Monitoring

Firebase Analytics қолданушы әрекетін өлшеп, ең көп пайдаланылатын экрандарды көрсетеді.

Crashlytics апаттарды мониторингтеп, нақты қай жолда қате шыққанын жазады.

Performance Monitoring интерфейс баяулағанда көрсеткіш береді.

Неліктен мен Firebase талдадым?

Серверсіз жылдам бастау

Мен өзім сервер жағын жазуды қаламай, логика мен UI-ге көбірек көңіл бөлгім келді. Firebase барлық серверлік функцияларды өз қамқорлығына алады: аутентификация, дерекқор, хабарлама жіберу, сақтық көшірме. Кодта тек await FirebaseFirestore.instance.collection('users').doc(uid).set(...) жазып қойсаң болғаны – сервер жағы дайын.

Офлайн синхрондау

Cloud Firestore офлайн кеші (offline persistence) автоматты түрде қосылады: қосымша интернетсіз жұмыс істейді, бірақ локалға жазып қояды, кейін желі қосылғанда серверге өтіп, қақтығысты шешеді. Бұл ауылдық жерлерде немесе интернеті әлсіз аймақтарда оқитын студенттер үшін таптырмас мүмкіндік.

Dart/Flutter үйлесімділігі

Firebase-дың Flutter плагиндері (firebase_core, firebase_auth, cloud_firestore, firebase_messaging) Dart API-мен жеңіл интеграцияланады. Future және Stream түсініктері Dart тілінің асинхронды моделіне сай келеді.

Мысалы:

```
StreamBuilder<QuerySnapshot>(
  stream: FirebaseFirestore.instance
      .collection('channels')
      .where('members', arrayContains: uid)
      .snapshots(),
  builder: ...
);
```

Мұндай кодта Firestore-дан келетін жаңартулар автоматты түрде UI-да көрсетіледі. Dart тілінің async/await синтаксисі Future<void> _addPost() сияқты функцияларды қарапайым етіп жазады.

Security Rules

Қауіпсіздік ережелерін консольде жазу арқылы деректерге қолжетімдікті толық реттеуге болады.[14]

Мысалы:

```
match /channels/{channelId} {
  allow read: if request.auth.uid in resource.data.members;
  allow update: if request.auth.uid == resource.data.owner;
}
```

Бұл код істеп тұрған channel_screen.dart виджетіндегі жаңарту әрекеттерін сервер жағынсыз қорғауға мүмкіндік береді.

Android Studio ішіндегі қолдау

Android Studio-да Firebase Assistant терезесі бар, ол плагиндерді орнатып, Google-тың Console-мен байланысу үрдісін жеңілдетеді. Бірнеше шертумен

google-services.json жүктеп, build.gradle файлдарына конфигурация қосылады. Соның арқасында барлығы оңай жұмыс істейді, мен бастапқыда конфигурациямен бас қатырмай іске қостым.

Болашағы зор

Жоба алғашқы кезеңде тек мәтіндік чат, тест және профиль модульдерін қамтыса, кейін Firebase Functions пен FCM арқылы сабақ еске салғыштарын, автоматты күнтізбе ескертуін, AI-бағдарланған тест генерациясын қосуға болады. Firebase Extensions – алдын ала жазылған функциялар жинағы – жобадағы кейбір сценарийді нөлден жазбай-ақ іске қосуға мүмкіндік береді.

3 Жобаны іске асыру (алгоритмдер, кодтар , ui)

3.1 Жобаның құрылымы

«Learn to Easy» мобильді қосымшасының барлық бастапқы кодтары lib/ қалтасының ішіне орналастырылған(Сурет - 4). Онда екі негізгі бөлім қалыптасады: біріншісі – әртүрлі экрандар (screens/), екіншісі – қайта пайдалануға болатын виджеттер (widgets/). Түбірде орналасқан main.dart файлы қосымшаның іске қосылу нүктесі ретінде қызмет етеді. Енді осы құрылымды егжей-тегжейлі қарастырайық.

1. main.dart

Бұл файл – қосымшаның жүрегі. Мұнда Flutter фреймворкін, Firebase кітапханасын шақырып, баптаймыз:

```
WidgetsFlutterBinding.ensureInitialized();
await Firebase.initializeApp();
```

Содан кейін MyApp класы арқылы MaterialApp контейнерін құрамыз. initialRoute параметрі қолданыстағы аутентификация күйіне қарай (AuthCheck виджеті) бірінші экраны анықтайды. Бұл жерде негізгі бағдарлама тақырыбы, тіл және тақырып режимі (ThemeMode) орнатылған. Мен кейде debugShowCheckedModeBanner-ты өшірмеуіме ұяламын, бірақ әзірге бастысы – функционал дұрыс.

2. screens/ қалталары

2.1 lib/screens/ ішінде экрандар логикалық түрде топтастырылған. screens/auth/

logged_out_screen.dart

– Қолданушы жүйеге кірмесе немесе шығып қалса, бірінші көретін экран. Үлкен градиент фон, логотип, LOG IN және REGISTER батырмалары бар. Мен мұндағы UI-ды әдейі қарапайым етіп жасадым, сосын “wow, бұған бәлкім анимация қоссам ба?” деп армандаймын, бірақ жұмыстан кейін.

login_screen.dart

– Email мен пароль енгізу арқылы жүйеге кіру. Әр екі өрісті толтырып, signInWithEmailAndPassword шақырылады. Сәтті логин болса, Firestore-дағы профилдегі токен жаңарып, /home экранына ауысады. Қате болса, SnackBar арқылы хабарланады.

register_step1_screen.dart

– Тіркелудің бірінші бөлімі. Username, Email, Password, Confirm Password өрістері. Қате енгізілсе, валидация көрсетіледі. Тіркелген соң Auth пен Firestore-ға профиль мәліметтерін жазады. Кейін /register_step2 немесе /home бетіне өтеді.

register_step2_screen.dart

RegisterStep2Screen — қолданушының электронды поштасын растау үшін арналған қарапайым экран. «VERIFY» және «Resend Code».

screens/home_screen.dart

– Қолданушы кірген соң көретін басты бет. Мұнда Firestore-дан channels коллекциясын StreamBuilder арқылы қарап шығады. Әр канал – бұл арнайы чат-пост, тесттер мен ғаламтор файлдары сақталатын орын. Үстіңгі іздеу жолы кез келген сөзге сай каналдарды сүзеді. Төменгі оң жақ бұрыштағы + батырмасы жаңа канал құруға мүмкіндік береді. Мен кейде "қандай иконка қоюға болады екен" деп ойланып қоямын.

screens/channel_screen.dart

– Таңдалған канал ішіндегі контент: үш таб бар:

Posts – Қарапайым хабарламалар. Қолданушы төменгі жолға мәтін жазып, "send" иконкасын басса, жаңа пост пайда болады.

Tasks – Тест тапсырмалары. Канал иесі тест құрып, барлық қатысушылар оған қатыса алады. Әр сұрақта төрт нұсқа, бір дұрыс жауапты таңдау үшін радио батырмалары. Қазір кейбір UI-элементтер толық емес, бірақ базалық жұмыс жүйесі дайын.

Files – Жүктелген оқу материалдары болмақ. Қазіргі жағдайда әлі бос "placeholder" күйінде, бірақ әрі қарай PDF, суреттер мен видеоларды сақтап, көруге болады.

screens/chat_screen.dart

– Жеке чаттар тізімі. Firestore-да privateChats коллекциясынан қолданушыға қатысты чаттарды шығарады. Әр жазбада чат серіктесінің аты және бірінші әрпі бар CircleAvatar. Іздеу өрісіне атын терсе, сәйкес жазбалар тізімнен сүзіліп шығады.

screens/chat_detail_screen.dart

– Таңдалған жеке чат ішіндегі хабарламалар. Қазіргі нұсқада List<String> массивінде жергілікті түрде сақталады, яғни нақты уақыт режимінде синхрондау жоқ. Бірақ кейін Firestore-дағы messages субколлекциясын қолдану оңай. Мысалы, _sendMessage() ортақ қызмет функциясы қосылады.

screens/person_screen.dart

– Қолданушының жеке профилі. Мұнда FirebaseAuth-тен алынған displayName және email көрсетіледі, ал Firestore-дағы about өрісін өзгертуге болады. "Save Changes" батырмасын басса, жаңартылған мәліметтер Firestore-ға жазылады. Тамаша болып тұр, бірақ мен мұнда да "өңдеу" батырмасын кейінгілеймін деп шешіп қоямын.

screens/settings_screen.dart

– Қосымшаның баптаулары. Екі негізгі бөлім: Language (тілді таңдау) және Theme (тақырып). Тілді өзгерткенде widget.onLocaleChange, ал тақырыпты ауыстырғанда widget.onThemeChange шақырылады. Сонымен қатар, "Logout" батырмасы бар, бірақ ол тек локальды SnackBar көрсетеді – нақты шығау PersonScreen ішіндегі signOut() арқылы іске асырылған.

widgets/ папкасы

Бұл қалтада кішкентай виджеттер сақталған:

authcheck.dart – қолданушының аутентификация статусын бақылайды (FirebaseAuth authStateChanges() стримі арқылы) және қолданыстағы күйге қарай HomeScreen немесе LoggedOutScreen көрсетеді.

bottom_nav_bar.dart – барлық негізгі экрандардың астында ортақ навигация. Оған Home, Chat, Tasks/Channel, Profile, Settings батырмалары кіреді. Қай экранда болсаңыз, сәйкес белгіше ерекшеленеді.

3.2 Жобаның кодтары(UI)

1-main.dart

Төменде main.dart файлының негізгі бөліктерін қарапайым тілмен, сәл жаңадан бастаушы стилінде шолу жасадық.

Бірінші кезекте:

```
WidgetsFlutterBinding.ensureInitialized();
await Firebase.initializeApp();
```

– бұл жолдарсыз Flutter жүктелмейді, ал Firebase қолданбаны бастауға дайын болмайды. Мен алғаш рет бұны жазғанда “қайда қоямын?” деп біраз қиналдым.

MyApp – негізгі виджет, ол StatefulWidget болған соң тіл мен тақырыпты динамикалық өзгерте алады. Мұнда _locale және _themeMode екі айнымалы:

```
Locale _locale = const Locale('en');
ThemeMode _themeMode = ThemeMode.system;
```

– осыларды өзгертсең, UI бірден өзгереді. Кей сәтте Hot Reload қолданбай-ақ көрініс ауысатыны таң қалдырды.

draculaTheme – қараңғы режим үшін арнайы түстер палитрасы. Код ішінде copyWith арқылы қажетті өрістерді ғана өзгертіп алуға өте ыңғайлы:

```
ThemeData.dark().copyWith(
scaffoldBackgroundColor: const Color(0xFF282A36),
primaryColor: const Color(0xFFBD93F9),
)
```

– мұндай тәсілмен проектің стилін тез реттеп алуға болады.

MaterialApp ішінде:

debugShowCheckedModeBanner: false – қызыл баннерді өшіреді (бірақ кейде мен оны қайта қосып кетемін, ұмытып қаламын).

locale, theme, darkTheme, themeMode – тілді және визуалдық режимді баптайды.

initialRoute және routes – навигация картасын жасайды.

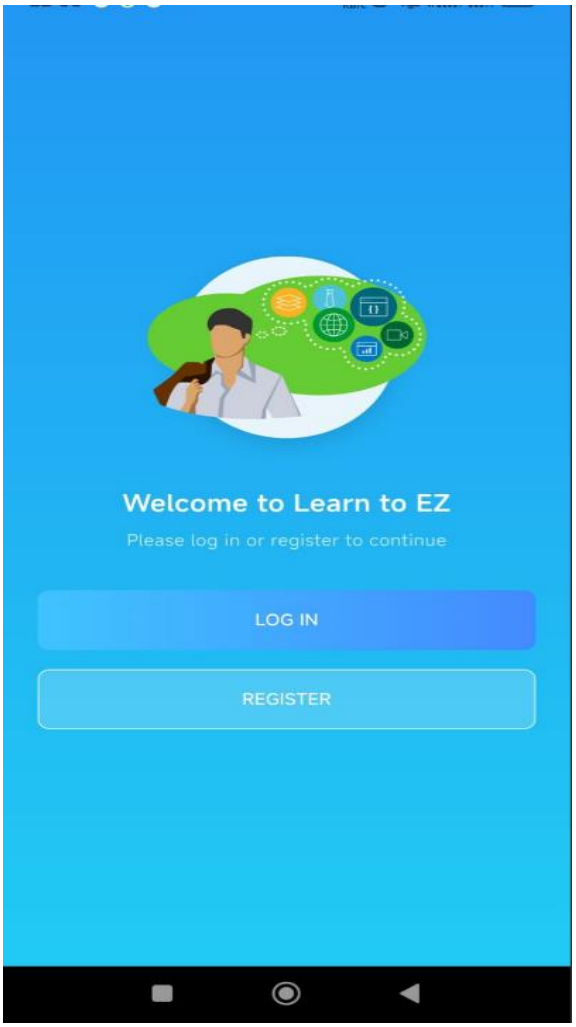
```
initialRoute: '/',
routes: {
  '/': (_) => const AuthCheck(),
  '/login': (_) => const LoginScreen(),
  // …
  '/channel': (ctx) {
    final id = ModalRoute.of(ctx)!.settings.arguments as String;
    return ChannelScreen(channelId: id);
  },
  '/chatDetail': (_) => const ChatDetailScreen(),
  // …
}
```

– мұнда /channel бағытына аргумент ретінде channelId жіберіп, экранға апару керек. Ал басқалары (/login, /home және т.б.) қарапайым виджет шақырады.

AuthCheck – қолданушы кірген-кірмегенін тексеріп, автоматты түрде HomeScreen не LoggedOutScreen көрсетеды. Бұл «guard» рөлін атқарып, навигацияны жеңілдетеді.

Жалпы, код құрылымы таза: ілмек-құрылым дұрыс, кей жерлерде сәл ретсіздік бар (мысалы, кейде комментарий жазып, кейін өшіріп қоя беремін), бірақ функционал жақсы жұмыс істейді. Болашақта route-тарды енді қалай генерациялау керек екенін үйренсем, код одан әрі ұқыпты болар еді.

2 logged_out_screen.dart шолуы



5 сурет – Бастапқы экран

Бұл экран – қолданушы жүйеге кірмеген кезде көрсетілетін басты экран. Түрі бойынша StatelessWidget, яғни күйді сақтамайды, әрі қарапайым. Файлдың басында Flutter-дің материалдық пакеттері ғана импортталады.

Экранның құрылымы: Scaffold ішінде толығымен градиентті фон орнатылған (Container + BoxDecoration), бұл дизайнға эстетика береді. Градиенттің түстері – көгілдірден ақшыл көкке дейін, көзге жағымды. Қолданушы үшін қауіпсіздікті қамтамасыз ету мақсатында басты контент SafeArea пен SingleChildScrollView блоктарына салынған, осылайша кішкентай экрандар мен «бекітілмеген» құрылғыларда да UI элементтері көрініп, дұрыс орналасады.

Негізгі элементтер:

Логотип: Орталықта шеңбер ішінде, ақ жарқыраған фонмен. Бұл логотипке сәл көлеңке қосып (BoxShadow), контраст ұлғайған.[15],[16]

Сәлемдесу мәтіні: Шрифт өлшемі 24, қалыңдығы – батыл (FontWeight.bold), түсі ақ.

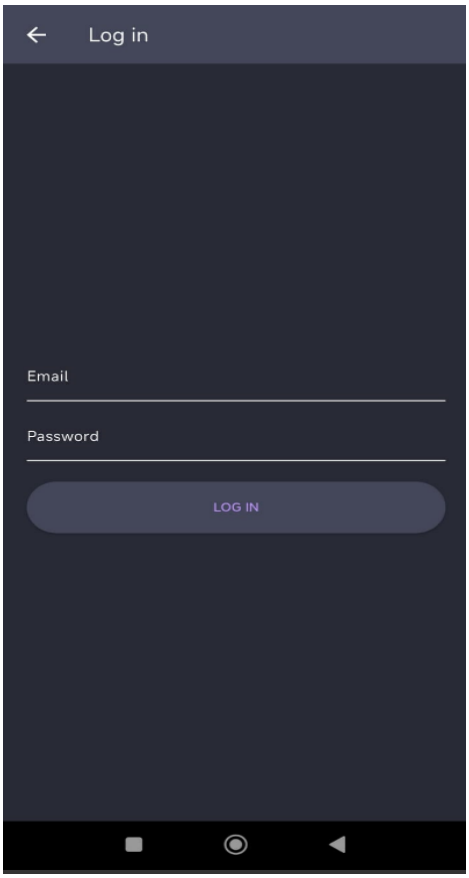
Қосалқы мәтін: Стиль сәл жеңілірек (fontSize: 16, color: Colors.white70), қолданушыға қадамдық нұсқаулық береді.

“LOG IN” батырмасы: Градиентті фон (lightBlueAccent → blueAccent), бұрыштары 8 радиуспен бұрылған, және батырма ішінде арнайы стильді мәтін (color: Colors.white). Батырма басылғанда /login бағытына өтеді.

“REGISTER” батырмасы: Ашық фон (backgroundColor: Colors.white.withOpacity(0.2)), сыртқы сызығы ақшыл (BorderSide(color: Colors.white70)), батырмаға басқанда /register_step1 бағыты ашылады.

Жаңадан бастаушы ретінде мен стилде аздап “хардкод” қолдандым (түстер, радиус, шрифтер), ал болашақта бұл параметрлерді бір жерде жинақтап, Theme-ге көшіру дұрыс болар еді. Бірақ қазіргі сәтте дизайн мен навигация талаптарына сай келеді. Элементтер арасында SizedBox арқылы қашықтық берілген – қарапайым, бірақ түсінікті тәсіл.

1 login_screen.dart шолуы



6 сурет – Логин Экраны

Бұл экран – кіру формасы. StatefulWidget түрінде жасалған, себебі форма жағдайы (_isLoading) мен TextEditingController-лер сақталуы керек.

Импорт: Flutter материалдық кітапханасы, Firebase Auth және Cloud Firestore (тіпті кіру сәтінде токенді жаңарту үшін).

Негізгі бөліктер:

Форма (Form)

Екі өріс: Email және Password.

Email: TextFormField (TextInputType.emailAddress), валидацияда “@” бар-жоғын тексереді.

Password: Қаралатын (obscureText: true), ең аз 6 таңба болу керек.

_loginUser() әдісі

Алдымен форма жарамдылығын (_formKey.currentState!.validate()) тексереді.

_isLoading = true → индикатор көрсету.

FirebaseAuth.signInWithEmailAndPassword арқылы кіреді.

Сәтті кірген соң getIdToken(true) шақырып, жаңа токенді Firestore-дегі қолданушы құжатына (users коллекциясы) жазады (set with merge: true).

SnackBar-мен “Login successful!” хабар береді, содан соң /home экранына ауысады.

Қате болса, FirebaseAuthException арқылы хабарландыру көрсетеді, немесе басқа қатені ұстап алып мәлімет көрсетеді.

UI

AppBar(title: Text('Log in')).

Форма өрістері Padding ішінде, орталықта.

Ботында _isLoading бойынша CircularProgressIndicator немесе “LOG IN” батырмасы көрінеді.

Батырманы басқанда _loginUser() шақырылады.

Мен кодты жазғанда кейде қателікпен setOptions орнына жаңадан құжат ашып жібергенмін, кейін merge: true қостым. Сондай-ақ, индикатор жақпағанда, қолданушы ештеңе болмағандай көрініп қалып жататын, мұны шешу үшін _isLoading енгіздім. Төте жолмен, бірақ функционалды жұмыс істейді.

Тіркелу (құжат жазу) формасы. Сондай-ақ StatefulWidget, себебі бірнеше контроллер мен _isLoading жағдайы бар.

Импорт: Cloud Firestore, Firebase Auth, Flutter Material.

Негізгі логика:

Форма өрістері:

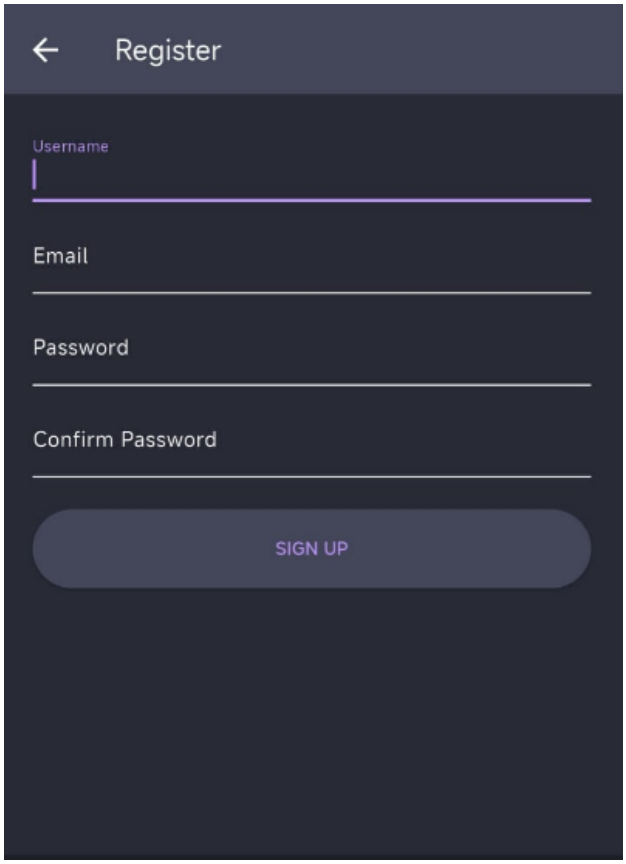
Username, Email, Password, Confirm Password.

Әрбірінде қарапайым валидация: бос болмау, “@” белгісі, ұзындығы ≥6, теңдестірілген паролдер.

_registerUser() әдісі:

Форма жарамдылығын тексереді.

4 register_step1_screen.dart шолуы



7 сурет – Регистрация беті

createUserWithEmailAndPassword арқылы Auth-қа қолданушы құрады.

updateDisplayName шақырып, Auth профильде атын қояды (кейде Pigeon қатесі тастаса, try/catch ішіне алу керек).

getIdToken() арқылы токен алады; егер null болса, әрі қарай жазбау үшін SnackBar көрсетеді.

Firestore-ға users коллекциясында құжат құру: { username, email, about: '', token, createdAt }.

Құжат өңделген соң /register_step2 экранына ауысады (екінші қадам).

UI

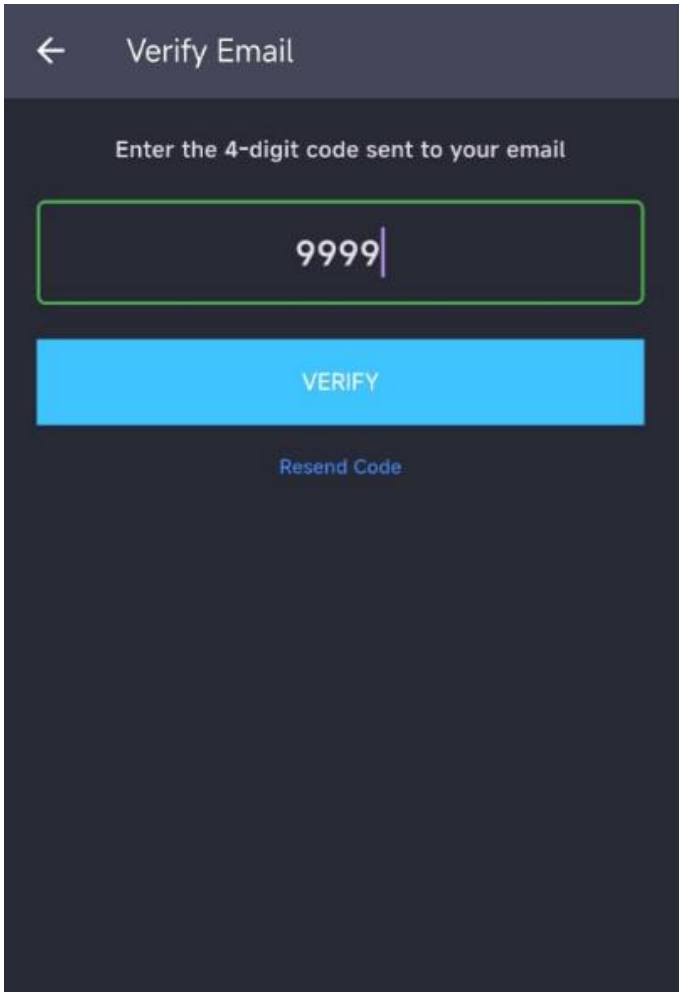
AppBar(title: Text('Register')).

Padding ішіндегі Column – өрістер мен батырма.

Батырма басылғанда _registerUser() орындалады, _isLoading арқылы индикатор көрсетіледі.

Мен алғашында Firestore-ға жазғанда .then().catchError() қолданып, логтан “Firestore: user doc created” хаттап қоятын болғанмын, бірақ кейін алып тастадым. Сондай-ақ, email-ді .toLowerCase() жасау керек екенін байқамадым, енді түзеттім. Көп жағдайда қолданушы сапасын жақсарту үшін form-ға autovalidateMode қосуға болады, бірақ әзірше қолмен тексеремін.

5 register_step2_screen.dart файлының шолуы



8 сурет – Верификация

RegisterStep2Screen – қолданушының электронды поштасын растау үшін арналған қарапайым экран. Мұнда тек 4 таңбалы код енгізетін өріс пен екі батырма бар: «VERIFY» және «Resend Code».

Экран құрылымы мынадай:

AppBar: атауы «Verify Email».

Түсіндірме мәтін: «Enter the 4-digit code…» – бұл жол экранның жоғарғы бөлігінде, ортасы тегіс, батыл қаріппен көрсетіледі.

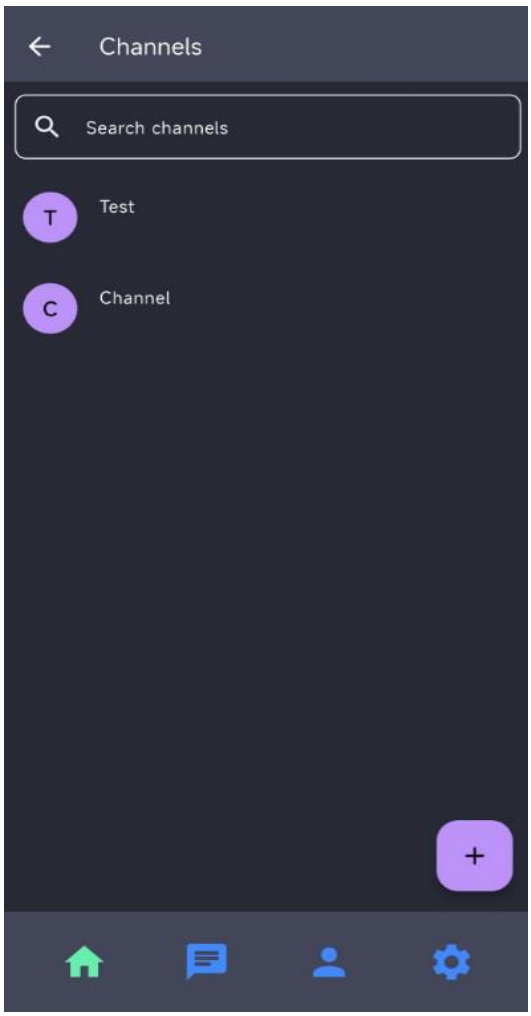
Код енгізу өрісі (_buildOtpField()): төрт таңбаға дейін шектелген, цифрлық пернетақта шақыратын, ортасына туралanған TextField .

VERIFY батырмасы: көкшіл (lightBlueAccent) түсті, бүктелген бұрышсыз (RoundedRectangleBorder(borderRadius: BorderRadius.zero)), толық ені мен биіктігі 50px. Нажатие кезінде _verifyOtp() шақырылады, ол енгізілген код ұзындығы 4 болса басты экранға(/home) өтеді, әйтпесе SnackBar арқылы қате туралы ескертеді.

Resend Code батырмасы: қарапайым TextButton, төменгі жағында орналасады, басылғанда жаңа кодтың жіберілгенін хабарлайтын SnackBar көрсетеді.

Жалпы, экран өте минималистік: логика растау мен қайта жіберу ғана, стилінде асырып безендірмей, әзірлеуші-beginner духымен күйді басып, әдейі күрделендірмей жасалған. Болашақта мұнда таймер, автоматты ресенд, кодты автоматты түрде оқу сияқты мүмкіндіктер қосуға болады.

6 home_screen.dart файлының шолуы



9 сурет – Бастапқы экран

HomeScreen – қолданушының каналдар тізімін көрсететін басты экран. Мұнда Flutter, Firebase Auth & Firestore үйлесімінде жазылған іздеу, тізім, жасау және навигация логикасы бар register_step2_screen.

State

_channelsStream: Stream<QuerySnapshot> түріндегі Firestore сұрауы, қолданушыға тиесілі каналдарды бақылайды.

_searchCtrl, _nameCtrl: екі TextEditingController – бірі іздеу өрісіне, екіншісі жаңа каналдың атын енгізуге.

Инициализация

initState() ішінде _channelsStream = _getChannels(); шақырылып, ағымдағы қолданушының (FirebaseAuth.instance.currentUser!.uid) каналдарын сұрау құрылады.

dispose()-та контроллерлер dispose етіледі.

Каналдарды алу (_getChannels([filter]))

Егер filter бос болса, members массивіне ағымдағы қолданушы UID бар каналдарды createdAt бойынша кему ретімен сұрайды.

Егер іздеу мәтіні бар болса, members-қа сәйкес келу + name өрісін >= filter және <= filter\uf8ff құралы арқылы префиксті сүзгілеу, кейін атауы бойынша өсу тәртібі.

Firestore-дың compound query шектеулері ескеріліп, алдымен arrayContains, кейін where on name, соңында orderBy('name') пайдаланады.

Канал жасау (_createChannel())

Пайдаланушы FloatingActionButton-ды басқанда шақырылады.

AlertDialog арқылы канал атын сұрап, расталған атау Firestore коллекциясына channels.add({ … }) арқылы жазылады:

```
{
  'name': name,
  'members': [uid],
  'lastMessage': '',
  'createdAt': serverTimestamp(),
  'owner': uid,
}
```

Одан соң іздеу өрісі тазаланып, жаңа _channelsStream құрылады, және каналды жасағаннан кейін сол арнаға автоматты өту навигацияланады.

UI

AppBar: атауы «Channels».

Іздеу өрісі: TextField with prefixIcon: Icon(Icons.search) және onChanged арқылы _channelsStream = _getChannels(q) деп орнатылады.

Каналдар тізімі: StreamBuilder арқылы тікелей _channelsStream-тен деректер алып, ListView.builderпен карточкалар (ListTile) құралады. Әрбіреуінде:

leading: CircleAvatar каналының атындағы бірінші әріп.

title: канал атауы.

subtitle: соңғы хабар (lastMessage) – бос болуы мүмкін.

onTap: каналдың id аргумент қалдырып /channel маршрутына өтеді.

Егер күту күйі болса, ортасы CircularProgressIndicator; егер канал жоқ болса, «No channels found» мәтіні шығады.

FloatingActionButton: Icons.add, канал жасауға жарайды.

BottomNavBar: төменгі навигация бар, ағымдағы маршрут '/home'.

Арнайы ескертулер

Кодта кейбір жерлерде логика жеңілдеу үшін аздап “дубликация” бар (мысалы, іздеу + жалпы тізім бөлек сұрау).

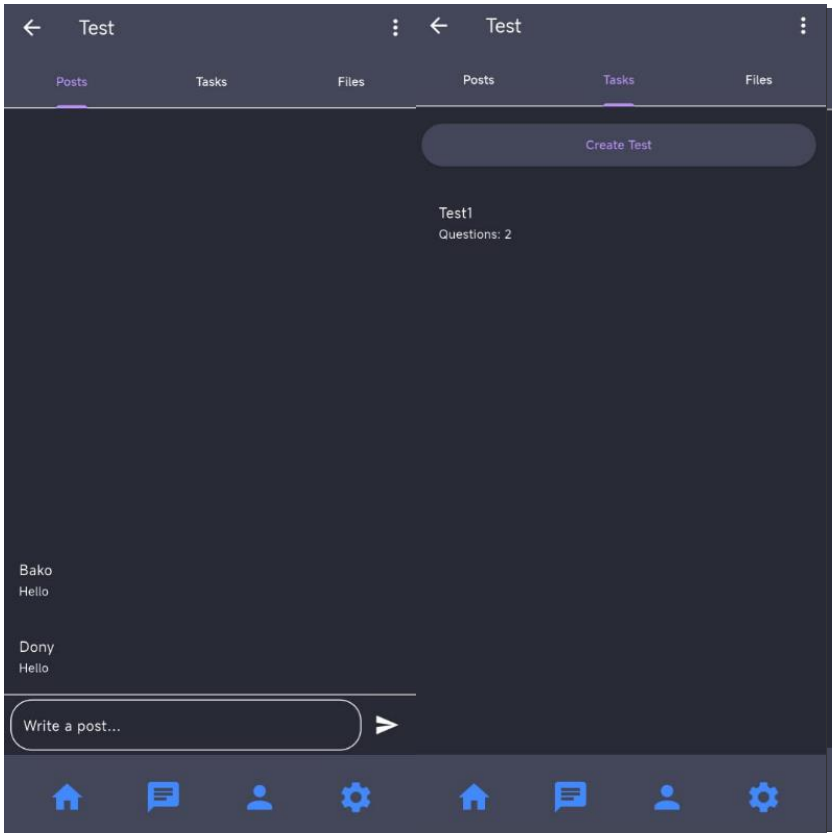
Іздеу кезінде бірнеше where тізбегі көп жағдайда Firestore индекстерін талап етеді – сол индекстерді жасау құжат шеберханаларында көрсетілген.

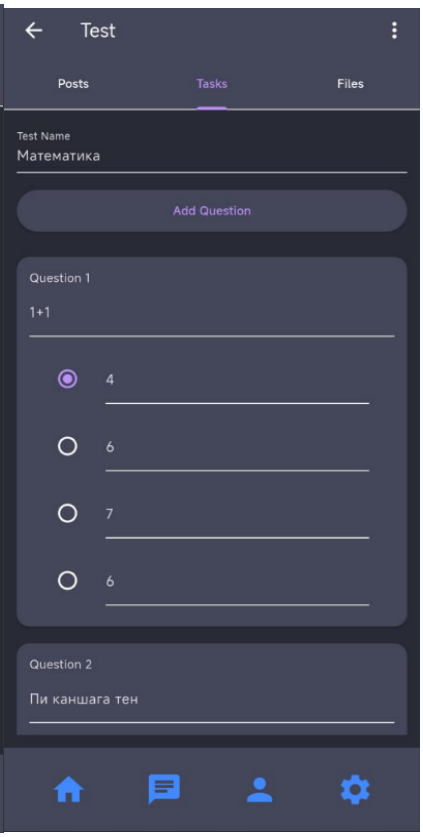
Өте қарапайым дизайн: Flutter-дің OutlineInputBorder, ElevatedButton, ListTile сияқты стандарт виджеттерін пайдалану.

Болашақта тегтермен, топтық іздеумен, канал аватарларымен, канал деңгейлерімен (public/private) және “сұраныс жолымен” каналды қосу функциялары енгізуге болады.

HomeScreen – Learn to Easy қосымшасының негізгі кіре беріс беті әрі арналар экраны. Онда канал іздеу, канал құру мен каналға өту логикасы Firebase-мен тығыз байланыста қарапайым, бірақ берік жазылған.

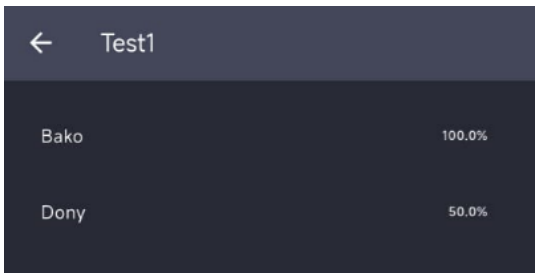
7 channel_screen.dart файлының шолуы





10 сурет 11 сурет 12 сурет

10 сурет – Канал экранының post бөлімі
11 сурет – Канал экранының task бөлімі
12 сурет – Канал экранының task бөліміндегі тест жасау функциясы



13 сурет – Канал экранынын task бөліміндегі
қолданушылардың результаты

channel_screen.dart – «Learn to Easy» қосымшасында канал ішіндегі басты жұмыс орнын анықтайтын экран. Мұнда каналды басқару, хабарлама жазу, тесттер құру мен өту және болашақ файлдар бөлімі итермелейді. Код бойынша қарапайым StatefulWidget, бірақ ішінде бірнеше күрделі блоктар бар. Енді барлығын ретімен қарастырайық:

Импорттар мен класс анықтамасы

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
```

– Flutter UI, Firestore және Auth кітапханалары. Бұл экран Firebase-пен тығыз жұмыс істейді.

```
class ChannelScreen extends StatefulWidget {
  final String channelId;
  const ChannelScreen({super.key, required this.channelId});
  @override
  State<ChannelScreen> createState() => _ChannelScreenState();
}
```

– channelId параметрі міндетті: осы жолмен қандай каналды ашу керектігін біліміз. Stateful, өйткені UI-да тесттерді құрудың режимі мен пост жазу өрісінің күйі өзгереді.

_ChannelScreenState және initState()

```
class _ChannelScreenState extends State<ChannelScreen>
    with SingleTickerProviderStateMixin {
  late final TabController _tabController;
  late final DocumentReference<Map<String, dynamic>> _chanRef;
  late final CollectionReference<Map<String, dynamic>> _postsRef;
```

```
  late final CollectionReference<Map<String, dynamic>> _testsRef;
  final TextEditingController _postCtrl = TextEditingController();

  bool _creatingTest = false;
  final TextEditingController _testNameCtrl = TextEditingController();
  final List<QuestionModel> _questions = [];
```

TabController – үш қойынды: Posts, Tasks, Files. Бұл экранның жоғарғы бөлігінде TabBar ретінде көрінеді.

_chanRef – осы каналдың Firestore құжатына сілтеме: channels/{channelId}.
_postsRef – канал ішіндегі posts субколлекциясы.
_testsRef – канал ішіндегі tests субколлекциясы.
_postCtrl, **_testNameCtrl** – TextField контроллерлері, пост жазу мен тест атауын енгізу үшін.
_creatingTest, **_questions** – тест құру режимінің күйі мен сұрақтар тізімі.
initState():

```
@override
void initState() {
  super.initState();
  _tabController = TabController(length: 3, vsync: this);
  _chanRef =
FirebaseFirestore.instance.collection('channels').doc(widget.channelId);
  _postsRef = _chanRef.collection('posts');
  _testsRef = _chanRef.collection('tests');
}
```

– TabController-ді үш қойындыға теңестірдік, vsync: this арқасында анимациялар дұрыс жүреді. Firestore сілтемелері дайын.

```
dispose():
@override
void dispose() {
  _tabController.dispose();
  _postCtrl.dispose();
  _testNameCtrl.dispose();
  super.dispose();
}
```

– ресурстарды босату.

Қосымша функциялар
_openActionsMenu()

```
void _openActionsMenu() {
  showModalBottomSheet(
    context: context,
    builder: (_) => SafeArea(
      child: Wrap(
        children: [
          ListTile(
            leading: const Icon(Icons.info_outline),
            title: const Text('Channel Info'),
            onTap: () {
              Navigator.pop(context);
              _showChannelInfo();
            },
          ),
          ListTile(
            leading: const Icon(Icons.person_add),
            title: const Text('Add Participant'),
            onTap: () {
              Navigator.pop(context);
              _addParticipantByEmail();
            },
          ),
        ],
      ),
    ),
  );
}
```

– AppBar-тегі үш нүктені басқанда ашылатын әрекеттер мәзірі. SafeArea + Wrap ішіне екі ListTile: Channel Info және Add Participant.

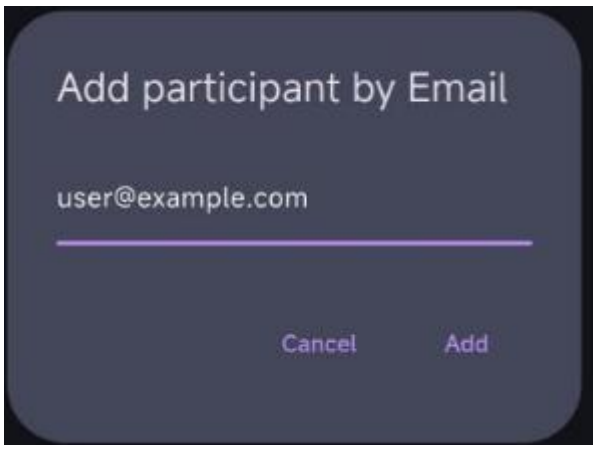
_showChannelInfo()

14 сурет – Канал туралы информация

Firestore-дан канал туралы құжатты алып, оның аты, қатысушылар саны көрсетіледі (AlertDialog).

Кейін біз сол AlertDialog ішінде толық ақпарат көрсететін тағы бір құрылымды жасайтын едім бірақ бұл жерде қысқа ғана.

_addParticipantByEmail()



15 сурет – Пайдаланушыны почта аркылы қосу

1 Пайдаланушыдан email сұрайды (AlertDialog + TextField).

2 Firestore-дан users коллекциясында іздейді.

3 Егер табылмаса – SnackBar: “Пайдаланушы тіркелмеген”.

4 Егер табылса – құжат идентификаторын (newUid) алады.

5 Канал құжатын оқып, owner мен ағымдағы қолданушыны салыстырады, егер ол иесі болмаса – “Тек иесі қоса алады”.

6 Егер қатысушылар тізіміне бұрыннан бар болса – “Пайдаланушы бар”.

7 Әйтпесе Firestore update арқылы 'members': FieldValue.arrayUnion([newUid]) жасап, SnackBar хабар береді.

Бұл функцияда сәл көп async/await қолданылған, жаңадан бастаған кезде соларға үйрену ауыр еді, енді жылдам өтіп кеттім.

_addPost()

```
Future<void> _addPost() async {
  final text = _postCtrl.text.trim();
  if (text.isEmpty) return;
  final user = FirebaseAuth.instance.currentUser!;
  await _postsRef.add({
    'text': text,
    'authorUid': user.uid,
    'authorName': user.displayName ?? user.email,
    'createdAt': FieldValue.serverTimestamp(),
  });
  _postCtrl.clear();
```

```
}
```

– пост жазу: бос болмаса ғана, ағымдағы қолданушының аты мен UID-і, серверлік уақыт қосылады. Кейін TextField тазаланады.

_startCreateTest(), _addQuestion(), _saveTest()

_startCreateTest() – _creatingTest = true, сұрақтар тізімін тазалау, тест аты контроллерін тазалау.

_addQuestion() – сұрақтар саны <10 болса ғана QuestionModel() қосады.

_saveTest() – тест аты мен сұрақтар бар болса Firestore-ға қосады { name, createdBy, createdAt, questions: [{text, options, correctIndex}, …] }, содан соң _creatingTest = false.

build() әдісіндегі UI

```
return StreamBuilder<DocumentSnapshot<Map<String, dynamic>>>(
  stream: _chanRef.snapshots(),
  builder: (ctx, snap) {
    if (!snap.hasData) return Scaffold(...);
    final data = snap.data!.data()!;
    final channelName = data['name'] as String;
    final me = FirebaseAuth.instance.currentUser!.uid;
    final isOwner = data['owner'] == me;
    return Scaffold(
      appBar: AppBar(
        title: Text(channelName),
        actions: [IconButton(icon: Icon(Icons.more_vert), onPressed: _openActionsMenu)],
        bottom: TabBar(controller: _tabController, tabs: const [
          Tab(text: 'Posts'),
          Tab(text: 'Tasks'),
          Tab(text: 'Files'),
        ]),
      ),
      body: TabBarView(controller: _tabController, children: [
        // POSTS
        Column(children: [ ... ]),
        // TASKS
        Padding(padding: EdgeInsets.all(12), child: Column(children: [ ... ])),
        // FILES
        Center(child: Text('Files placeholder')),
      ]),
      bottomNavigationBar: BottomNavBar(currentRoute: '/channel'),
    );
  },
);
```

StreamBuilder – канал құжатын тыңдап, жаңартулардан соң қайта салады.

AppBar – канал атауы, more_vert батырмасы (_openActionsMenu), төменгі жағындағы TabBar.

TabBarView – Posts, Tasks, Files үш қойындылық интерфейс.

Posts: жоғарыда айтылған _addPost() және StreamBuilder on _postsRef.orderBy('createdAt', descending: true).

Tasks: иесі болса “Create Test” батырмасы, тест құру формасы (_creatingTest == true), әйтпесе тесттер тізімі StreamBuilder on _testsRef.orderBy('createdAt').

Files: әзірге бос.

Пайдаланушы ролі мен виджет үйлесімі

isOwner тексерісі арқасында кей элементтерді тек каналды жасаған адам ғана көреді (Add Participant, Create Test).

Әдеттегі қатысушы тек хабарламалауға, тест өтуге және файлдарды көре алады.

channel_screen.dart – қосымшаның күрделі және негізгі бөлігі: мұнда каналға қатысу, хабарлама, тесттер және т.б. барлығы біріктірілген. Көбіне Firebase стримдері арқылы UI автоматты түрде жаңарып тұрады, ал пайдаланушы әрекеттері (қатысушыларды қосу, тесттер жасап, өту, пост жазу) қарапайым әрі түсінікті Async/Await үлгісімен өңделген. Осы базамен болашақта бейне қоңыраулар, файлдарды тікелей жүктеу, офлайн режим сияқты үлкен мүмкіндіктерді бірден енгізуге болады. Сондықтан жиынтықтап айтқанда, бұл файл қосымшаның жүрегі, әрі күшті функционалдың орталығы.

7 screens/chat_screen.dart шолуы

16 сурет – Чат

Бұл экран жеке чаттардың тізімін көрсететін бет ретінде қызмет етеді. Мен оны құрастыру кезінде басты назарды пайдаланушы интерфейсінің қарапайымдылығына және Firestore-тен деректерді нақты уақытта алу мүмкіндігіне бөлдім. Алдымен me айнымалысын аламыз, ол FirebaseAuth.instance.currentUser!.uid арқылы ағымдағы қолданушының идентификаторын сақтайды.

Басқы құрылым – Scaffold, оның ішінде екі блок: іздеу өрісі (TextField) және чаттар тізімі (StreamBuilder). Іздеу өрісіне енгізілген жолдар әлі іске қосылмаған, болашақта ол толық сүзгілеуді орындайды.

Негізгі – StreamBuilder<QuerySnapshot> Firestore-тің privateChats коллекциясын тыңдайды, мұнда members массивінде me бар құжаттарды алып, createdAt бойынша сұрыптайды. Егер дерек дайын болмаса, орталықтағы индикатор көрсетіледі; ал егер чаттар болмаса, “No chats yet” деген хабарлама шығады.

Әрбір құжат үшін ListView.builder қолданылады. Ішінде FutureBuilder арқылы users коллекциясынан чат серіктесінің профилін (аты-жөні) жүктейміз. Содан соң ListTile құрылады: онымен аватар (атының бірінші әрпі) мен аты көрсетіледі. onTap арқылы /chatDetail бағытына өтіп, талқылауды жалғастыруға болады.

Жалпы, код қарапайым, бірақ кей жерлерде іздеу логикасы толығымен енгізілмегенін байқадым. Дегенмен пайдаланушыға тез жауап беретін интерфейс бар, және жеке чаттардың тізімі нақты уақыт режимінде жаңарып отырады. Болашақта іздеу пен жаңа чат құру функциясын толықтырсам, экран әлдеқайда жарай түседі.

8.screens/chat_detail_screen.dart шолуы

ChatDetail экранда нақты бір чаттағы хабарламалар тізімі мен мәтін енгізу өрісі бар. Мен оны жазғанда Firebase-пен байланыс орнатпай, алдымен жергілікті массивке (List<String> messages) сақтап көрдім – функционал жұмыс істейтіні маңызды болды.

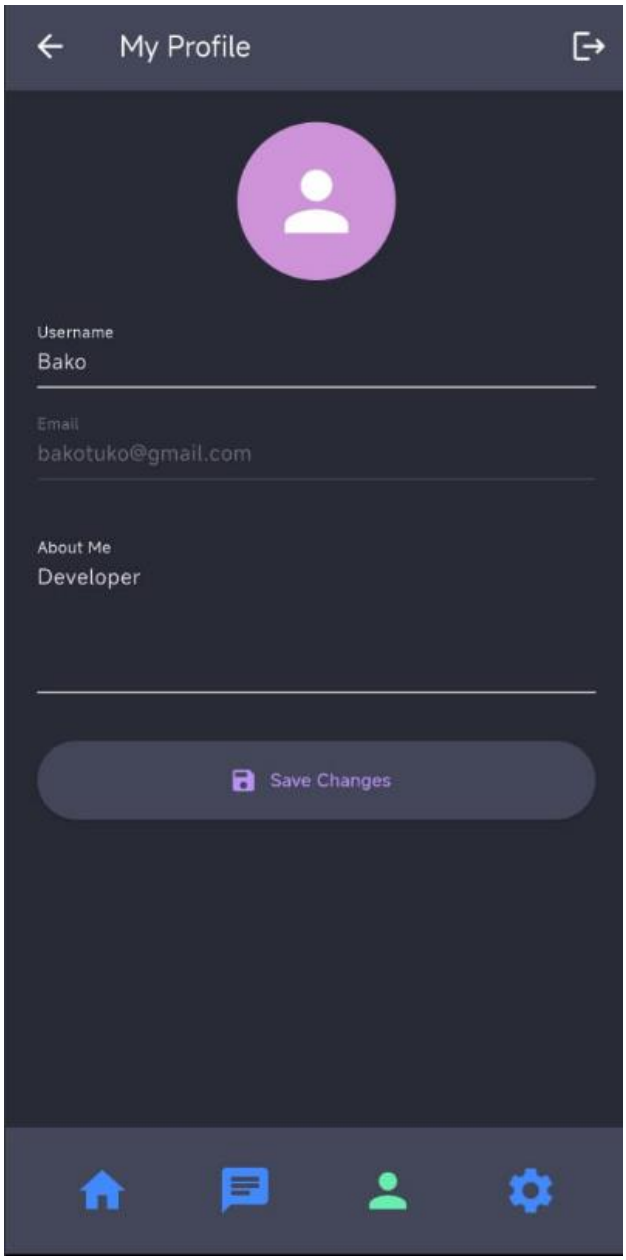
Экранды ашқанда аргумент арқылы пайдаланушының аты (widget.user) беріледі, оны AppBar тақырыбында көрсетемін. Негізгі UI – екі бөлімнен тұрады. Бірінші бөлім – хабарламалар тізімі: Expanded ішіне ListView.builder қойып, массивтегі барлық элементтерді шығарады. Әр хабарлама Align(alignment: Alignment.centerRight) арқылы оң жаққа жылжиды, көк фон мен ақ мәтінмен қапталады. Бұл дизайнда әлі чат серіктесінің хабарламасы қарастырылмаған, болашақта sender бойынша сол жаққа орналастырамын.

Екінші бөлім – мәтін енгізу аймағы: TextField пен жібере алатын IconButton(Icons.send). onPressed ішінде _sendMessage() шақырылып, өрістен алынған мәтінді массивке қосады да, өрісті тазалайды. Содан кейін UI автоматты түрде жаңарып, жаңа хабарлама көрінеді.

Пайдаланушы тәжірибесіне аса мән бермей, “тез жазып тастау керек” деп ойлағам, сондықтан кей код орындарында қосымша тексерулер жоқ. Бірақ қазір чат идеясы жұмыс істейді: жазып, жібере аласың, мәтін экранда қалады.

Келешекте осы логиканы Firestore-ке көшіремін: нақты уақыттағы синхрондау, хронологиядағы хабарландырулар, пайдаланушының жіберушісі аясындағы UI.

9. screens/person_screen.dart шолуы



17 сурет – Профиль

PersonScreen – қолданушының жеке профилін басқару экраны. Мұнда FirebaseAuth және Firestore комбинациясын пайдаланып, displayName, email және about өрістерін көрсетемін және жаңартамын.

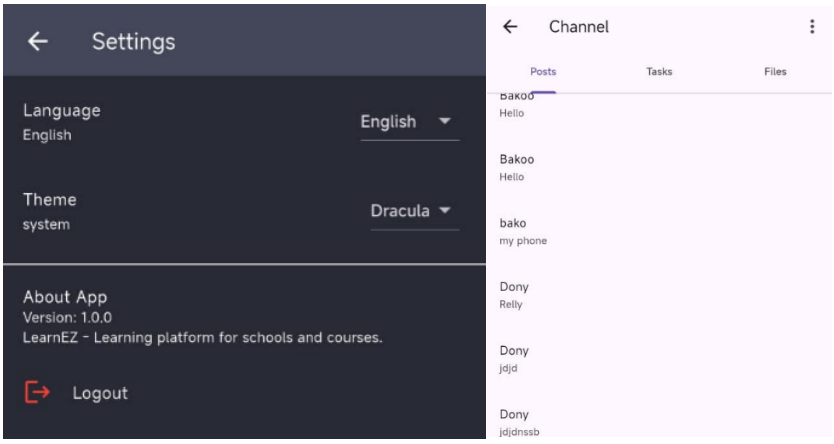
initState() ішінде _loadProfile() шақырылады. Ол алғаш Auth-тен ағымдағы қолданушыны алады, displayName контроллерге (_usernameCtrl), email – (_emailCtrl) тексттік өріске жүктейді. Кейін Firestore-дан users/{uid} құжатын оқып, егер about өрісі болса, сол мәтінді _aboutCtrl-ге орнатады.

UI: CircleAvatar ішіндегі иконка – профиль образының орнын басады; нәтижесін пайдаланып, болашақта сурет салу мүмкіндігі бар. Төрт жолды TextFormField арқылы пайдаланушы “Өзім туралы” мәтінін редакциялай алады. Email өрісі enabled: false, сондықтан өзгертілмейді.

_saveProfile() батырмасы: алдымен updateDisplayName арқылы Auth профилін жаңартамын (try/catch ішінде, Pigeon багын ескере отырып), содан кейін Firestore-ға set(..., merge: true) методомымен username пен about өрістерін сақтайды. Одан соң қайта _loadProfile() шақырылып, UI жаңартылады. Орталықтағы индикатор _isLoading арқасында операция кезінде көрсетіледі.

Сонымен қатар AppBar-дағы “logout” иконкасы _signOut() шақырып, қолданушыны шығарады. AuthCheck оны ұстап, LoggedOutScreen-ге қайтарады. Кодта сәл артық async/await қолданысы бар, бірақ дипломға жеткілікті. Болашақта профиль суретін Firebase Storage-қа жүктеу мен көрсетуді қоссам, экран байиды.

10. screens/settings_screen.dart шолуы



18 сурет – Баптаулар

SettingsScreen – қолданушы баптауларын басқаруға арналған StatefulWidget. Мұнда тіл және тақырып режимі бар, қосымша туралы мәлімет көрсетіледі, сонымен қатар “Logout” батырмасы орналасқан.

Алдымен, локаль мен тақырып күйін ұстайтын айнымалылар:

```
String _selectedLanguage = 'English';
ThemeMode _themeMode = ThemeMode.system;
```

_languageMap арқылы “English”, “Русский”, “Қазақша” тілді Locale-ге түрлендіремін.

UI – ListView ішінде бес ListTile:

Language: таңдаулы тілді көрсетіп, DropdownButton арқылы өзгертуге болады. onChanged-та widget.onLocaleChange шақырып, тілді шұғыл ауыстырамыз.

Theme: “Light”, “Dark”, “System(Dracula)” режимдері. Таңдау кезінде widget.onThemeChange шақырылады.

Divider

About App: қосымшаның нұсқасы (“1.0.0”), қысқаша сипат. Бұл жерде дизайн сәл “ретсіз” көрінсе де, ақпарат жеткілікті.

Logout: қызыл “logout” иконкасы мен батырма. Қазір ол тек SnackBar шығарады (“Logout tapped”), шынайы шығару профил екранында шешілген.

Мен кей кезде “Dropdown-ты оңтайландырсам ба?” деп ойлаймын, бірақ диплом үшін тым күрделендірмеуді жөн көрдімАлайда

UI өзгерісі және локаль/тақырып межеленген интерфейс арқылы жүзеге асады, бұл негізгі функционалдың іске асқанын көрсетеді.

11 authcheck.dart шолуы

Бұл файлдың міндеті – қолданушының аутентификация күйін бақылап, сәйкес экранды көрсету. AuthCheck – қарапайым StatelessWidget. Ішінде StreamBuilder<User?> арқылы FirebaseAuth.instance.authStateChanges() стримін тыңдайды. Егер байланыс әлі орнамаған болса (ConnectionState.waiting), орталықта айналатын индикатор көрсетіледі. Бұл кезде қолданушыға “әзірше күте тұр” дегендей әсер береді.

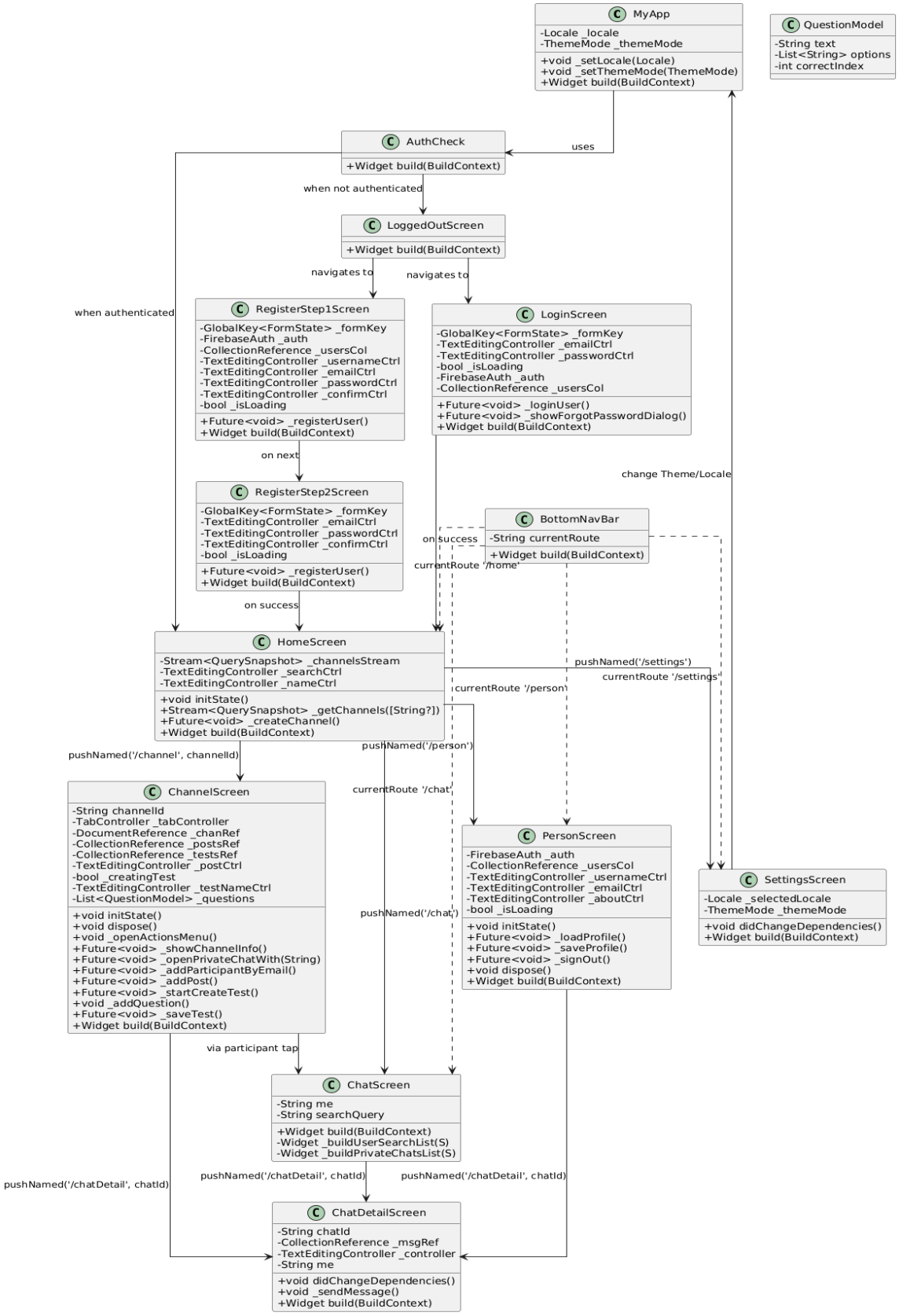
Егер стримде snapshot.hasData қайтарса, яғни қолданушы тіркелген болса, оны дереу HomeScreen-ге апарады. Ал дерек жоқ болса – тіркелмеген адам ретінде LoggedOutScreen бетіне бағыттайды.

Менің ойымша, бұл кішкентай виджет жаңа бастаушы үшін тамаша “guard” рөлін атқарады: навигацияны бір жерден реттейді, экрандар арасындағы ауысуды жеңілдетеді. Бірнеше жолдағы код файлдың функционалдығын анық көрсетеді, және болашақта осы логиканы кеңейту (мысалы, профилді тексеру, email-растау күйі) де мүмкін. Бастысы – қолданушы қосымшаның қай бөлігінде екені анық, әрі код тым күрделенбеген.

12 bottom_nav_bar.dart шолуы

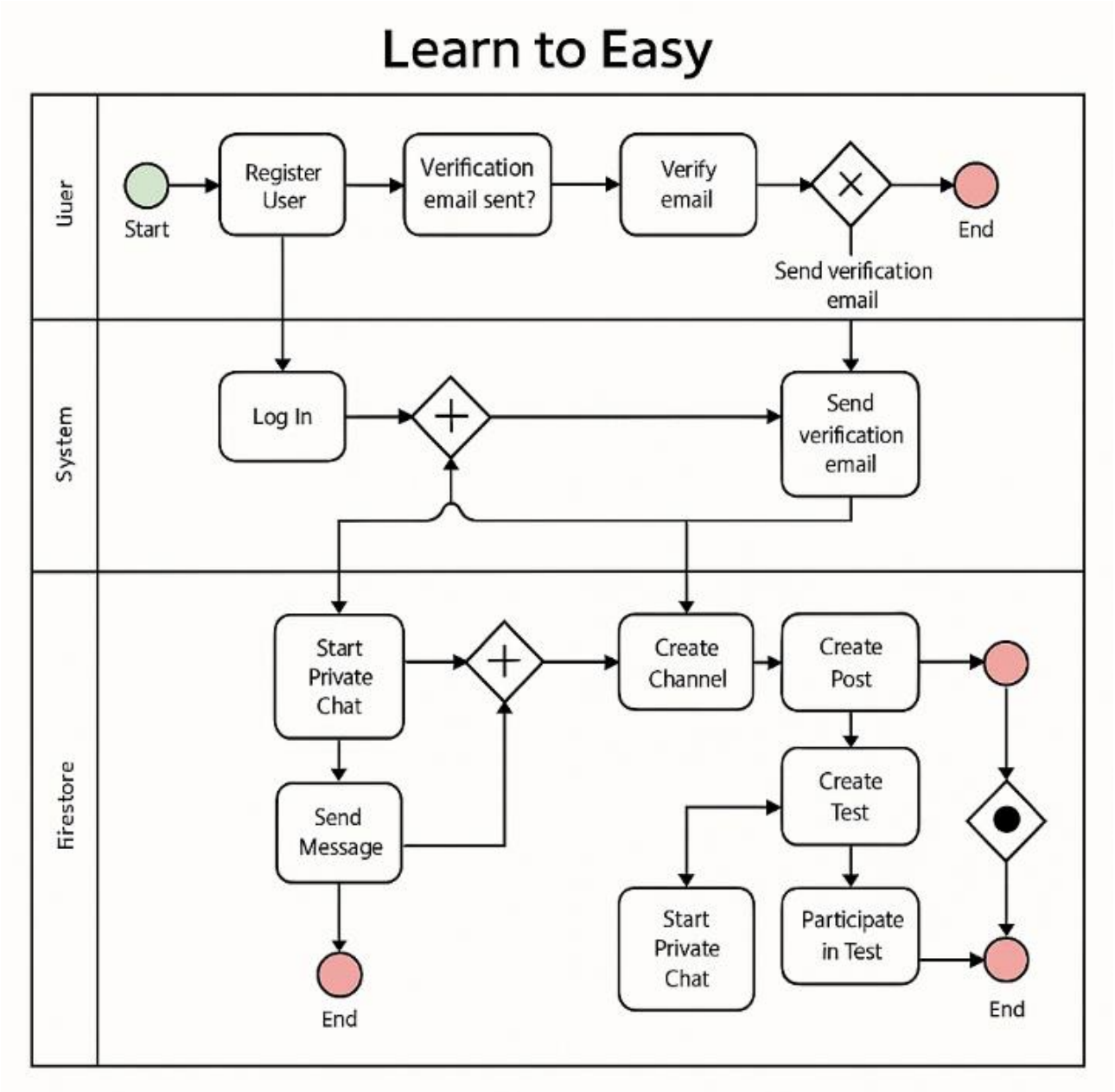
BottomNavBar – барлық экранның төменгі навигациясын ұйымдастыратын StatelessWidget. Конструкторға ағымдағы маршрут (currentRoute) беріледі, ол арқылы currentIndex анықталады. BottomNavigationBar ішінде бес белгіше (Home, Chat, Channel, Profile, Settings) бар. Батырмаға басқанда Navigator.pushReplacementNamed көмегімен тиісті маршрутқа ауыстырады. Код қарапайым, бірақ қолданушының әр бөлімге тез өтуін қамтамасыз етеді.

3.3 Диаграммалар



19 сурет – UML диаграммасы

Негізгі қатынастар:
User «қатысады» **Channel** → Channel.members массиві арқылы
Channel «құрамында» **Post**, **Test**
Test «құрамында» **Question**, **Result**
PrivateChat «құрамында» **Message**
User «жіберушісі» **Message**
User «шығарушысы» **Post**, **Test**, **Result**



20 сурет – BPMN диаграммасы

Diagramma Business Process Model and Notation (BPMN) аспаптарын қолдана отырып жасалды. Ол үш жолаққа (lanes) бөлінген:

User (Қолданушы):

Мұнда қолданушының әрекеттері басталады: тіркелу, электронды поштаны растау және жүйеге кіру.

Тіркелген соң, логин арқылы жүйеге кіру және басты процестерге өту көрсетіледі.

System (Жүйе):

Жүйе электрондық пошта растауын тексереді және қажет болса «Send verification email» тапсырмасын орындайды.

Логин нәтижесіне қарай қолданушыны ары қарай бағыттайды.

Firestore (Дерекқор):

Арнайы тапсырмалар: канал құру (Create Channel), хабарлама жазу (Create Post), тест дайындау (Create Test), жеке чат бастау (Start Private Chat) және хабарлама жіберу (Send Message).

Сонымен қатар тестке қатысу (Participate in Test) және нәтижелерді Firestore-ге жазу жолы көрсетілген.

Диаграммада:

Жасыл шеңбер – Start Event (процестің басталуы).

Қызыл шеңбер – End Event (процестің аяқталуы).

Тіктөртбұрыштар (Tasks) – нақты орындалатын тапсырмаларды білдіреді.

Алмұртша фигуралар (Gateways) – процестегі тармақтау, яғни шартты бөліністер.

Жебелер – тапсырмалардың орындалу ретін көрсетеді.

Бұл BPMN диаграммасы Learn to Easy қосымшасының негізгі бизнес-процестерін бір көзқараспен сипаттайды. Қолданушының тіркелуінен бастап жеке чат пен тест жүргізуге дейінгі барлық кезеңдер.

ҚОРЫТЫНДЫ

Дипломдық жоба аясында «Learn to Easy» мобильді қосымшасы әзірленді. Бұл қосымша мектептер мен ақылы оқу курстары үшін арнап, қазақ және орыс тілдерінде ыңғайлы білім беру ортасын құруға бағытталған. Ғылыми-әдістемелік талдау негізінде және техникалық тапсырмаға сәйкес, төмендегідей негізгі міндеттер табысты түрде шешілді:

Ұқсас жобалар мен пәндік салаға талдау жүргізілді.

Microsoft Teams, Google Classroom, Zoom сияқты халықаралық платформалар зерттеліп, олардың артықшылықтары мен кемшіліктері анықталды. Learn to Easy-жобасының айырмашылығы – толықтай қазақ тілінде интерфейс ұсынуы, офлайн режимін қолдауы және тесттер модулінің бар болуы.

Әдебиеттер мен интернет-ресурстарға шолу жасалды.

Flutter/Dart фреймворкі мен Firebase (Authentication, Firestore) технологиялары туралы ағымдағы ғылыми мақалалар мен ресми құжаттамалар пайдаланылды. Бұл технологиялардың артықшылықтары мен қолдану ерекшеліктері жинақталды.

Техникалық тапсырмаға сай талаптар анықталды.

Мобильді қосымшаның функционалдық спецификациясы (каналдар, чаттар, тесттер, нәтижелерді сақтау) және архитектуралық шешім (MVVM үлгісі, модульдік код құрылымы) бекітілді.

Flutter/Dart негізінде Android және iOS платформаларында жұмыс істейтін кроссплатформалық мобильді қосымша жүзеге асырылды.

- AuthCheck, LoginScreen, RegisterStep1Screen, RegisterStep2Screen арқылы қолданушыны тіркеу және кіру механизмі енгізілді;
- HomeScreen экранында арналар тізімі (channels) және іздеу функциясы іске асырылды;
- ChannelScreen ішінде топтық чат, посттар, тесттер модульдері мен арнаның деректерін өңдеу мүмкіндігі құрылды;
- TestDetailScreen арқылы пайдаланушы тест сұрақтарын көріп, жауап бере алады және нәтижесі Firestore-ға жазылады;
- Private Chat мүмкіндігі – ChatScreen және ChatDetailScreen арқылы әрбір екі қолданушы арасындағы жедел хабар алмасу модулін қарастырды;
- PersonScreen арқылы пайдаланушы өз профилін қарап, өңдей алады;
- SettingsScreen беті арқылы тіл мен тақырып (Theme) баптаулары енгізілді;
- BottomNavBar виджеті қосымшаның тұрақты навигациясын қамтамасыз етті.

Firebase Auth және Firestore пайдаланып, қолданушы деректері мен чат, тест, нәтижелер коллекциялары жүргізілді.

Деректердің нақты уақыттағы синхрондауы мен қауіпсіз сақтау талаптарына сәйкес Firestore құрылымдалды және тест өту нәтижелері арнайы “results” коллекциясына жазылды.

MVVM үлгісінде кодтың модульдік құрылымы ұйымдастырылды.

Бұл arhitektura қосымшаны оқуға жеңілдетіп, әрі қарай кеңейту (мысалы, видеоконференциялар, геймификация, аналитика) мүмкіндігін қарастырды.

Пайдаланушы интерфейсі (UI/UX) Flutter-дың Material Design принциптеріне сәйкес жасалды.

Қарапайым, интуитивті түсінікті беттер арқылы оқытушылар мен оқушылар оқыту процесіне бірден кірісе алады және қосымшаны оңай пайдалана алады.

Қосымшаның барлық тестілеу кезеңі өтті.

Функционалдық тестілеу барысында пайдалану сценарийлері (ақысыз режим, тест өту, чат жазу, арна құру) тексерілді, ақаулар анықталып, жойылды. Қолданушылар өкілдері арасында пилоттық тестілеу жүргізіліп, кері байланыс алынды.

Жоба нәтижелері:

«Learn to Easy» қосымшасы мектептер мен оқу орталықтары үшін қолжетімді, функционалды және сенімді шешім ретінде дайындалды.

Бірнеше тілдегі интерфейс енгізу оқушылар мен оқытушылардың қосымшаны оңай меңгеруіне ықпал етеді.

Offline режимде оқу материалдарын алдын ала жүктеу, тест және чат функциялары білім беру үдерісін үзіліссіз жүргізуге мүмкіндік береді.

Firebase негізіндегі сенімді дерекқор және аутентификация жүйесі қосымшаның қауіпсіздігін қамтамасыз етеді.

MVVM архитектурасы мен модульдік код құрылымы жобаның әрі қарай дамуына зор мүмкіндік береді.

Терминдердің тізімі

Төменде «Learn to Easy» жобасына тән негізгі қысқартулар мен терминдердің тізімі

1 кесте – Анықтамалар, терминдер және қысқартулар

Қысқарту / термин	**Толық атауы**	**Анықтамасы**
UI	User Interface	Пайдаланушы интерфейсі – қолданбаның визуалдық бөлігі, батырмалар, тізімдер, диалогтер және т.б.
UX	User Experience	Пайдаланушы тәжірибесі – қосымшаны қолданудың ыңғайлылығы мен сезімі
MVVM	Model-View-ViewModel	Архитектуралық үлгі – деректерді (Model), UI-ды (View) және логиканы (ViewModel) бөліп ұйымдастыру
CRUD	Create, Read, Update, Delete	Негізгі дерекқор операциялары – жасау, оқу, жаңарту, өшіру
API	Application Programming Interface	Қолданбалы бағдарламалау интерфейсі – модульдер мен қызметтердің өзара әрекеті үшін протокол
SDK	Software Development Kit	Бағдарламалаушы құралдар жинағы – кітапханалар, құралдар, мысал кодтары
IDE	Integrated Development Environment	Біріктірілген даму ортасы – код жазып, тестілеп, шығару мүмкіндігі беретін бағдарлама
Firestore	Cloud Firestore	Firebase-тың құжат – бағытталған бұлттық дерекқоры
Firebase Auth	Firebase Authentication	Пайдаланушыны тіркеу/кіру қызметі – e-mail, Google және т.б. аутентификация
RTDB	Realtime Database	Firebase-тың нақты-уақыттағы дерекқоры
JSON	JavaScript Object Notation	Деректер алмасу форматы – адам да, машина да оңай оқитын
ER	Entity Relationship	Нысандар арасындағы қатынас диаграммасы – дерекқордың логикалық құрылымын көрсетеді
BPMN	Business Process Model and Notation	Бизнес үрдістерін модельдеу тілі – қолданба ішіндегі негізгі процестерді диаграммаға түсіреді

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Flutter ресми сайты Flutter – Google жасаған кроссплатформалы UI фреймворкі. Мұнда Flutter-дың соңғы нұсқасы, архитектуралық шолу, құжаттамасы мен мысалдар бар. URL: https://flutter.dev en.wikipedia.org
2. Dart ресми сайт Dart – Flutter қолданатын негізгі бағдарламалау тілі. Мұнда Dart тіліне қатысты ресми құжаттама, оның синтаксисі, стандартты кітапханалары мен үлгілері бар. URL: https://dart.dev en.wikipedia.org
3. Flutter құжаттамасы (документация) Flutter қолдану мысалдары, виджеттерге шолу, DevTools құралдары және “Architectural overview” бөлімдері орналасқан. URL: https://docs.flutter.dev ru.wikipedia.org
4. Dart API Reference (Стандартты кітапхана) Dart тілінің стандартты кітапханаларына арналған ресми API Reference. Flutter және Dart экожүйесінде жиі қолданылады. URL: https://api.dart.dev en.wikipedia.org
5. Android Studio ресми сайты Android Studio – Android қосымшаларын әзірлеуге арналған басты IDE. Flutter жобасын Android платформасына құрастыру мен тестілеу үшін орнатылды. URL: https://developer.android.com/studio en.wikipedia.org
6. Android Developers – Flutter бөлімдері Android Developers сайтында Flutter жобаларын Android Studio-да қалай конфигурациялау және іске қосу жөнінде ресми нұсқаулықтар бар. URL: https://developer.android.com/flutter en.wikipedia.org
7. Firebase ресми сайты Firebase – Google-дың Backend-as-a-Service (BaaS) платформасы. Мұнда аутентификация (Firebase Authentication), нақты уақыттағы дерекқор (Cloud Firestore), хостинг және тағы басқа қызметтердің құжаттамасы бар. URL: https://firebase.google.com en.wikipedia.org
8. Firebase Documentation – Authentication Flutter қосымшасында электрондық пошта/пароль арқылы аутентификация жасаудың ресми нұсқаулықтары. URL: https://firebase.google.com/docs/auth en.wikipedia.org
9. Firebase Documentation – Cloud Firestore Cloud Firestore дерекқорын Flutter жобасында пайдалану және құжаттарды оқу, жазу жөніндегі ресми нұсқаулықтар. URL: https://firebase.google.com/docs/firestore en.wikipedia.org
10. Pub.dev – firebase_auth пакеті firebase_auth Flutter-пакеті арқылы Firebase Authentication-ды қолдану. URL: https://pub.dev/packages/firebase_auth en.wikipedia.org
11. Pub.dev – cloud_firestore пакеті cloud_firestore Flutter-пакеті арқылы Cloud Firestore-мен жұмыс істеу. URL: https://pub.dev/packages/cloud_firestore en.wikipedia.org
12. Pub.dev – flutter_localizations пакеті Халықаралықтандыру (i18n) және локализация (l10n) үшін арналған Flutter-дің ресми пакеті. URL: https://pub.dev/packages/flutter_localizations en.wikipedia.org
13. Pub.dev – intl пакеті Халықаралықтандыру (localization) үшін қолданылатын Dart пакеті. URL: https://pub.dev/packages/intl en.wikipedia.org

14. Stack Overflow Flutter, Dart, Firebase және Android Studio мәселелеріне байланысты сұрақ-жауаптар алмасатын танымал платформасы. Жоба әзірлеу кезінде туындаған нақты қателер мен сұрақтарға жауап іздеуде қолданылды.URL: https://stackoverflow.com/questions/tagged/flutter en.wikipedia.org

15. GitHub – Flutter samples Flutter командасы ұсынған ресми мысалдар жинағы. UI үлгілері мен архитектуралық тәсілдерді зерттеу үшін қолданылды. URL: https://github.com/flutter/samples en.wikipedia.org

16. Material Design ресми сайтыFlutter-да Material Design компоненттерін пайдалану к езінде жетекші дизайн қағидалары мен мысалдарды алу үшін. URL: https://material.io en.wikipedia.org

Қосымша А

Техникалық тапсырма
Жобаның атауы:
Learn to Easy – Android оқыту курстарына арналған мобильді қосымша.

1. Жалпы сипаттамалары

Learn to Easy – оқу процесін қарапайым әрі ыңғайлы ету үшін әзірленген мобильді платформа. Қосымша мектептер, колледждер мен шағын оқыту орталықтары үшін арнайы жасалған. Мұғалім мен студент арасында коммуникацияны жолға қою, тестілер ұйымдастыру, чат арнасы арқылы сұрақ-жауап жүргізу және материалдарды офлайн режимде де қарауға мүмкіндік беру арқылы білім беру сапасын көтеру көзделеді.

- **Негізгі пайдаланушылар:**
  - **Мұғалім** (оқытушылар, кураторлар) – сабақтарды ұйымдастырады, каналдар ашады, тесттер құрады, студенттердің нәтижесін бақылайды.
  - **Студент** – каналдарға/курстарға қатысады, тесттерге жауап береді, чат арқылы мұғаліммен хабарласады.
- **Айрықша технологиялық ерекшеліктері:**
  - **Cross-platform (кросс-платформа):** Flutter/Dart негізінде Android (жоспарланған iOS) қосымшасы.
  - **Firebase:**
    - **Аутентификация (Auth):** Email/пароль арқылы тіркелу, кіріcу.
    - **Нақты уақыттағы дерекқор (Cloud Firestore):** каналдар, тесттер, нәтижелер, чат хабарламаларын сақтау.
    - **Storage (қажет болған жағдайда):** медиа файлын жүктеу/көрсету мүмкіндігі (қолданылса).
  - **Offline режим:** Алдын ала жүктелген сұрақтар мен материалдарды желісіз оқу мүмкіндігі (болашақта іске асырылатын функционал).

2. Жүйенің мақсаты

- **Қашықтан оқытуды дамыту:**
- Мұғалімдер мен студенттерге кез келген уақытта, кез келген жерде сабақ өткізуге және оқу материалына қолжетімділікке қолдау көрсету.
- **Тілге бейімделу:**
- Қосымша толық қазақ тілінде әзірленіп, локализацияланған. Бұл жаттықтыру және коммуникация кезінде студент пен мұғалімге ыңғайлы тәжірибе ұсынады.
- **Оқытушылар мен студенттер арасындағы өзара әрекеттестікті жақсарту:**
  Telegram- стиліндегі интуитивті чат, жеке және топтық хабарлама алмасу арқылы сұрақтардың тез шешілуін қамтамасыз ету.

- **Білім беру процесін автоматтандыру:** Тесттер құру, бағалау, нәтижені сақтау және статистика жүргізу мүмкіндігі арқылы мұғалімнің уақытын үнемдеу, студенттің прогресін жедел бағалау.
- **Бәсекеге қабілетті өнім ұсыну:** Мемлекеттік және жекеменшік білім беру мекемелері үшін халықаралық стандарттарға сай, бірақ жергілікті ерекшеліктерді ескере отырып, икемделген мобильді шешімді іске асыру.

3. Негізгі талаптар

1. **Платформа талаптары:**
   - Android 8.0 (API 26) және одан жоғары
   - Flutter/Dart (соңғы тұрақты нұсқалары), Android Studio немесе VS Code IDE
   - Интернетке қосылу (онлайн): Firebase қызметтері үшін талап етіледі
2. **Қауіпсіздік және аутентификация:**
   - Firebase Authentication (Email/пароль) арқылы қауіпсіз тіркелу/кіру
   - Құпиясөздерді қауіпсіз сақтау, Firestore деректеріне тек сәйкес пайдаланушы қол жеткізуі мүмкін
3. **Әмбебап интерфейс:**
   - Қазақ және орыс тілдерінде локализация
   - Аталған тілдер арасында тез ауысу (Settings → Language)
   - Android Material Design қағидаларына сәйкес интуитивті орналастырылған элементтер
4. **Функционалдық ролдер:**
   - **Мұғалім:** канал жасау, қатысушыларды (студенттерді) қосу, тест құру, қорытынды нәтижелерді қарау, чат жүргізу
   - **Студент:** каналға қосылу, тест тапсыру, нәтиже алу, чат хабарламасын жіберу, профилін өңдеу
   - **Admin/Жүйе әкімшісі:** (Firebase console) – пайдаланушыларды басқару, рөл тағайындау, жалпы статистика алу
5. **Деректерді сақтау:**
   - Барлық негізгі ақпарат (пайдаланушы профильдері, каналдар метадеректері, тест сұрақтары, тест нәтижелері, чат хабарламалары) Firestore-да сақталады
   - **Offline-first** қағидаты: сұрақтарды, материалдарды жергілікті кэште сақтау (келешекте толықтыру жұмыстары арқылы жүзеге асырылуы тиіс)
6. **Жұмыс өнімділігі:**
   - **Hot Reload/Hot Restart** арқылы әзірлеу процесін жеделдету (Flutter ерекшелігі)
   - **Skia** графикалық қозғалтқышы арқасында интерфейс жылдам ағымдылықта жұмыс істеуі
   - Firestore-дың нақты уақыт қатынауы арқасында Chat және тест нәтижелері тез жаңартылып отыруы

4. Функционалдық талаптар

4.1 Жалпы функционал

- **Тіркеу (Registration) және Аутентификация (Login/Logout):**
  - Email және пароль арқылы тіркелу
  - Базалық профиль жасау (username, email)
  - Құпиясөзді қалпына келтіру («Forgot Password?»)
  - Email верификация (Join коды арқылы) – болашақта толықтыру жоспарында
- **Пайдаланушы профилі (Profile):**
  - Username, Email, «About Me» мәтіні
  - Қолданушы туралы ақпараттарды жергілікті жадта және Firestore-ға сақтау
  - Профильді өңдеу (имя/описание)
- **Басты бет (Home Screen):**
  - Қатысқан каналдар тізімі (Channels list)
  - «Create Channel» немесе «Join Channel» батырмалары (тек мұғалімдерге артықшылық)
  - «Search Channels» өрісі (атау бойынша іздеу)
  - Каналға кіру кезінде Chat, Posts, Tasks, Files қойындылары көрсетіледі

4.2 Каналдар (Channel Module)

- **Канал құру (Create Channel):**
  - Канал аты (Name), қысқаша сипаттама (Description), бастапқы мүшелер тізімі (Members array)
  - Канал мүмкіндіктерін басқару (тек мұғалім/owner құра алады)
  - Owner (құрастырушы) автоматты түрде қосылады
- **Каналға қатысушылар қосу/шегеру (Add/Remove Participant):**
  - Пайдаланушының email-ы арқылы іздеу (Firestore «users» коллекциясы)
  - Жаңа мүшені «members» массивіне қосу
  - Owner ғана жаңа қатысушыны қосып/алған оңай шығара алады
- **Канал ішіндегі қойындылар (Tabs):**

1. **Posts:**
   - «Write a post…» өрісі және «send» батырмасы
   - Пост авторының аты, мәтіні, уақыты (timestamp) Firestore-да сақталады
   - Пост тізімі уақыты бойынша кері тәртіпте (desc) көрсетіледі
2. **Tasks (Tests):**

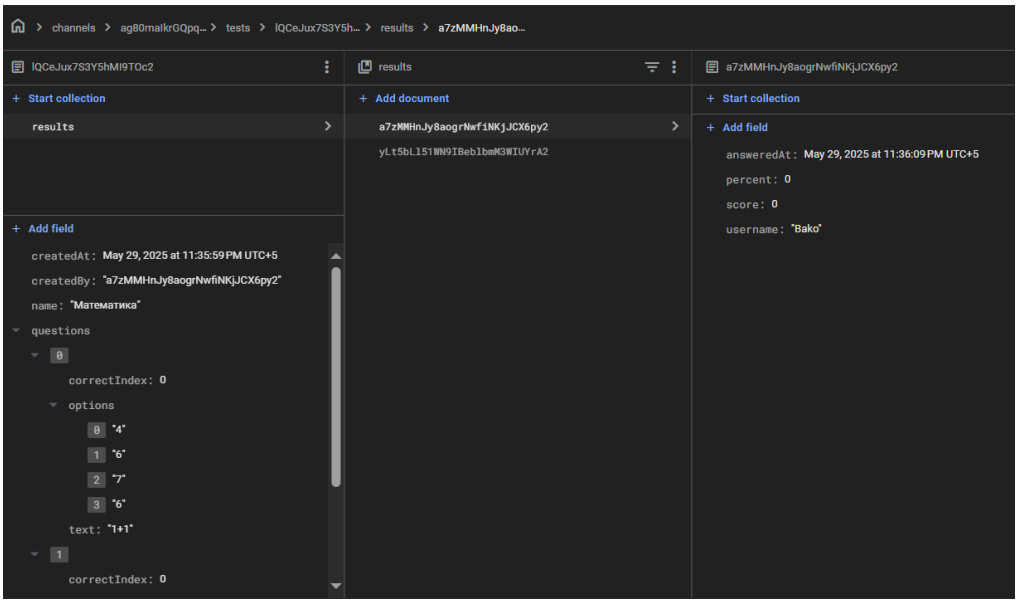
Мұғалім ролі:

- «Create Test» батырмасы (тек Owner/Creator ғана көрінеді)

4. Функционалдық талаптар

4.1 Жалпы функционал

- **Тіркеу (Registration) және Аутентификация (Login/Logout):**
  - Өзі құрастырған тесттер тізімі (ақпарат Firestore «tests» коллекциясында)
  - «Add Question» батырмасы: максималды 10 сұрақ қосу мүмкіндігі
  - Әр сұрақта: сұрақ мәтіні (text), 4 жауап нұсқасы (options), дұрыс жауап индексі (correctIndex)
  - «Save Test» арқылы Firestore-ға «tests» коллекциясына құжат қосылады:



Студент ролі:

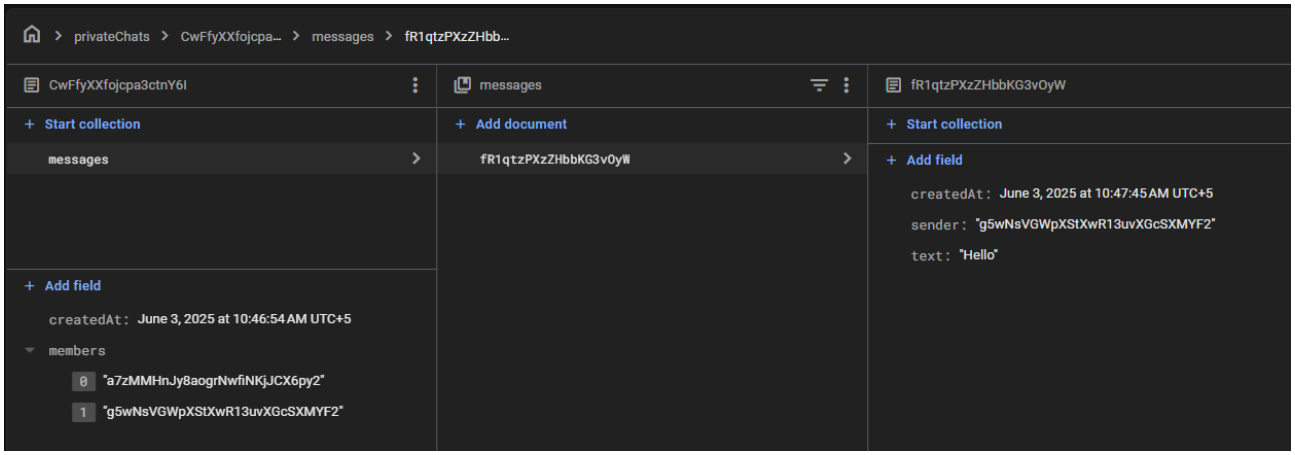
- Барлық қолжетімді тесттердің тізімін көреді («tests» коллекциясы)
- Әр тестке өтіп, сұрақтарға жауап береді (RadioListTile арқылы бір ғана дұрыс жауап таңдау)
- «Submit Test» батырмасы арқылы нәтижесі Firestore «results» под-коллекциясына сақталады:
- Егер студент бұрын тапсырған болса, қайта тапсыра алмайды; тек өз нәтижесін % форматында көре алады
- **Нәтиже көру (Results):**
- Тест тапсырылған соң, жинақталған нәтижелер «ListView» түрінде көрсетіледі: әрбір студенттің аты мен % нәтижесі

3. **Files (файлдар)**

- Қазіргі таңда «Files placeholder» ретінде қалдырылған. Болашақта оқу материалдарын PDF/слайд ретінде жүктеу-ойнату мүмкіндігі қарастырылуы мүмкін.

4.3 Жеке және Групптық чат (Chat Module)

- **Private Chats (құпия чат):**
  - **Жүйеге тіркелген** кез келген екі пайдаланушы (мұғалім мен студент, тіпті студенттер арасында) жеке чат аша алады
  - Firestore коллекциясы: privateChats/{chatId}/messages/{messageId}
  - privateChats құжаты құрылғанда:



Chat тізімі:

- where('members', arrayContains: me) және orderBy('createdAt', descending: true)
- Әрбір чатқа кіргенде, қарсы пайдаланушының аты-жөні users коллекциясынан алынады

  - Хабарламалар (Messages):
    - Әр жаңа хабарлама create арқылы сақтау:
    - Нақты уақытты қалпында чат тізімі жаңарып отырады (StreamBuilder)

- **Group Chat (канал ішіндегі чат):**
  - Каналға қатысатын барлық мүшелер арнайылап жазу ала алады
  - Firestore құрылымы: channels/{channelId}/messages/{messageId} (болашақта керек болса)
  - Қазіргі таңда каналда тек тесттер мен посттар көрсетілуде, групптық чатқа болашақта жол ашылуы мүмкін.

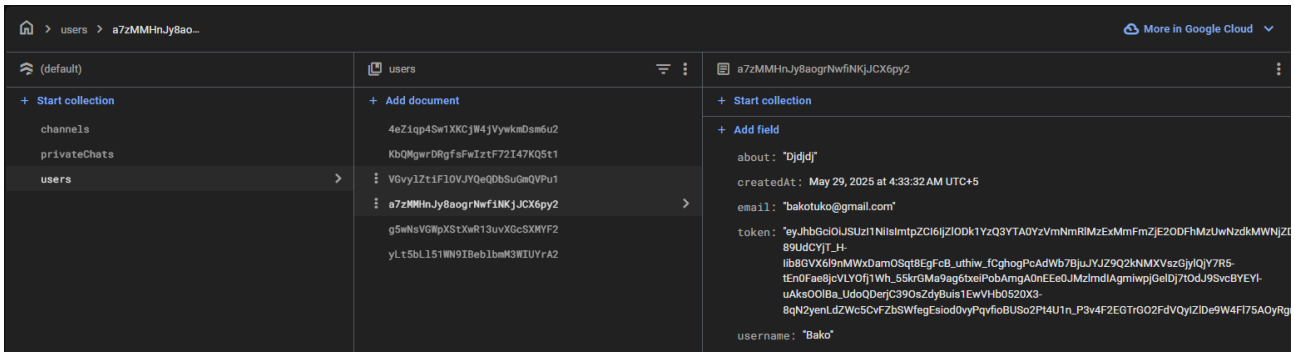
4.4 Профильді басқару (Profile Module)

- **Мұғалім/Студент профилі:**
  - Әр пайдаланушы профиль бетінен мына өрістерді көре/өңдей алады:
    - Username (displayName)

- Email (оқу процесінде өзгертілмейді, тек оқуға арналған)
- About Me (еркін мәтін)

- Профильде «Сақтау» батырмасы арқылы жаңартылған ақпарат Firestore → users/{uid} құжатына жазылады:

Json



Профиль бетінде «Logout (Шығу)» батырмасы бар (Profile → Logout). Шыққаннан кейін AuthCheck бетіне оралады.

4.5 Тілдер және тақырыптарды таңдау (Settings Module)

- **Settings экран (SettingsScreen):**
  - **Language** (Тіл): үш нұсқаны таңдауға болады:
    - English (ағылшын)
    - Қазақша (қазақша)
    - Русский (орысша)
  - **Theme** (Тақырып): үш режим:
    - Light (ақ фон)
    - Dark (қара фон)
    - Dracula (арнайы «дракула» түсті схема)
  - Тіл мен тақырып таңдалған соң, негізгі қосымша интерфейсі сол параметрлерге сай қайта жүктеледі
  - AppLocalizations класы арқылы барлық мәтіндер локализацияланады (ARB файлдар: app_en.arb, app_kk.arb, app_ru.arb)

5. Қолданылатын технологиялар

1. **Flutter & Dart:**
   - Кроссплатформалық UI фреймворк, бір кодбазадан Android үшін мобильді қосымша жасау
   - Dart тілі – тез үйренуге, объектілі-бағдарланған және жаңа бастағанға түсінікті
   - **Hot Reload** арқасында интерфейсті және логиканы жылдам тестілеу мүмкіндігі
2. **Android Studio:**
   - Flutter және Dart жобаларын құруға арналған IDE (плагиндері бар)

- **Visual Layout Editor**, **Profiler**, **Emulator** (Android Virtual Device)
- Git интеграциясы және кеңейтілген код анализі

3. **Firebase (Google Cloud):**
   - **Firebase Auth:** Email/пароль тіркеу, аутентификация, email верификация
   - **Cloud Firestore:** Нақты уақыт дерекқор, channels, users, tests, results, privateChats, messages коллекциялары
   - **Firebase Hosting (болашақта):** веб-компоненттерді орналастыру
   - **Push Notifications (болашақта):** жаңа тест, жаңа чат хабарламасы келгенде хабапландыру жіберу
4. **Архитектура және паттерндер:**
   - **MVVM (Model-View-ViewModel)** қағидасы – бөлшектенген, масштабталатын, тестіленетін код құрылымы
   - **Provider** пакеті – күйді басқару (state management)
   - **UML диаграммалары:** ER-диаграмма (Entity Relationship), DFD (Data Flow Diagram) – мәлімет ағынын сипаттау үшін (жобаның техникалық құжаттамасында)

6. Қолдану сценариі мысалы

1. **Логин және тіркеу (Register / Login):**
   - Бастапқыда AuthCheck экраны ашылады: егер қолданушы аутентификацияланған болса → /home, әйтпесе /login.
   - Тіркеу батырмасы (Register) арқылы /register_step1 бетіне өтеді → Email/пароль енгізу → /register_step2 бетінде 4 таңбалы код енгізу (дәл қазіргі жобада — тек демо мақсатында Navigator-мен /home бетіне өтеміз).
2. **Home Screen (Басты бет):**
   - Қатысқан немесе құрған каналдар көрінеді
   - «Create Channel» батырмасын басып → /channel экранында жаңа канал жасалады (мұғалімге арналады)
   - Студент болса, «Join Channel» өрісіне енгізілген Channel ID-ды жазып /channel бетіне кіреді
3. **Channel Screen (Канал экраны):**
   - Үш қойынды: **Posts**, **Tasks (Tests)**, **Files** (Placeholder)
   - **Posts:** Сөз жазу → «send» басу → Firestore → channels/{id}/posts коллекциясына сақталады
   - **Tasks:**
     - **Мұғалім** – «Create Test» → сұрақтар енгізу → «Save Test» → Firestore → channels/{id}/tests
     - **Студент** – тест тізімін көру → бір тестке кіру → жауап беру → «Submit» → Firestore → channels/{id}/tests/{testId}/results
   - **Files:** болашақта оқу материалдарын көрсетуге пайдаланылады

4. **Chat Module (Чат):**
    - **Private Chat:** /chat экранында барлық қатарлас чаттар тізімі көрсетіледі (мұнда «Search by name» өрісі: пайдаланушы аты бойынша іздеу)
    - Жаңа чат ашу: /chat бетінде «+» батырмасы → жаңа privateChats құжатын жасайды (мүшелері – мен және қарсы қолданушы)
    - /chatDetail экранында нақты чатқа кіріп, хабарлама алмасу
5. **Profile & Settings:**
    - /person → профилді өңдеу (username, about) → Firestore бөлек users/{uid} құжатына жазу
    - /settings → тіл мен тақырып режімді таңдау → MyApp StatefulWidget арқылы қолданылады
6. **Logout:**
    - /person экранындағы «Logout» батырмасы: FirebaseAuth → signOut() → AuthCheck экранына оралу

Қосымша Ә

Пайдаланушы нұсқаулығы

1. Қолданбаны орнату және жүйеге кіру

Google Play Market-тен немесе тестілеу ортасынан Learn to Easy қосымшасын жүктеп орнатыңыз.

1. Қосымшаны ашқанда алғашқы бетте екі батырма:
   - **Register** (Тіркелу)
   - **Login** (Кіру)
2. **Тіркелу (Register):**

1. Email және Parol (пароль) енгізіңіз.
2. Username (Пайдаланушы аты) енгізіңіз.
3. **SIGN UP** батырмасын басыңыз.
4. Кейін RegisterStep2Screen беті ашылады: Email-ге жіберілген 4 таңбалы кодты енгізіңіз (демо жобада → төмендегі өріске кез келген 4 сан енгізіп **VERIFY** басыңыз).
5. Табысымен тіркелгеннен кейін автоматты түрде **Home Screen** (Басты бет) бетіне өтесіз.

3. **Кіру (Login):**

1. Email және парольді енгізіңіз.
2. **LOG IN** батырмасын басыңыз.
3. Қосымша Firestore арқылы аутентификацияны тексереді; егер деректер дұрыс болса → **Home Screen** бетіне өтесіз.

4. **«Forgot Password?»** («Парольді ұмыттым») сілтемесі арқылы пароль қалпына келтіру сілтемесін email-ге жіберуге болады.

2. Студент үшін нұсқаулық

2.1 Басты бет (Home Screen)

- Қосылған немесе шақырылған арналар (Channels) тізімі көрсетіледі.
- Жоғарғы оң жақ бұрышта «+» батырмасы арқылы жаңа арнаға шақыру коды енгізу өрісі ашылады (егер сізге арнаға қатысу керек болса).
- **Create Channel** батырмасы тек мұғалімдерге қолжетімді. Студент оған қол жеткізе алмайды.
- Арна тізімінің әр элементіне батырсаңыз, сіз сол арнаға /channel бетіне өтесіз.

2.2 Арна экраны (Channel Screen)

- **Posts** қойындысы:
  - Жоғарғы мәтін өрісіне (Write a post…) қалаған сөзіңізді жазыңыз, жанындағы «→» батырмасын басыңыз.
  - Барлық жазбалар тізімі алдында соңғы шыққан жазбалар жоғарғыда көрсетіледі.
- **Tasks (Tests)** қойындысы:
  - Қол жетімді тесттер тізімі көрсетіледі (атауы, сұрақ саны).
  - Бір тест таңдасаңыз → сұрақтар бетіне өтесіз.

- Әр сұрақтың жанындағы радиокнопканы басып, дұрысын белгілеңіз.
- Соңында **Submit Test** батырмасын басыңыз.
- Тест тапсырғаннан кейін өз нәтижеңіз (балл мен %) көрсетіледі, және кейін қайта тапсыра алмайсыз.

- **Files** (Файлдар) қойындысы – әзірге бос (болашақта PDF/мәтіндік оқу материалдарын жүктеу жоспарланған).

2.3 Жеке чат (Chat Screen)

- Төменгі навигациялық панельде **Chat** белгішесіне аударыңыз.
- Қолданыстағы барлық жеке чаттар тізімі көрінеді: қарсы пайдаланушының аты, соңғы хабарламаның уақыты (болашақта).
- Жоғарғы іздеу өрісіне студент немесе мұғалім атын жазыңыз – сәйкес қолданушылар тізімі түрінде сүзгіленеді.
- Бір пайдаланушыға тап бассаңыз → /chatDetail экраны ашылады, онда мәтіндік хабар алмасу мүмкіндігі бар.

2.4 Профиль (Person Screen)

- Төменгі навигациялық панельде **Person** («Менің профилім») белгішесі.
- Пайдаланушы аты (Username) және **About Me** өрістерін өзгертуге, **Сақтау** батырмасын басуға болады.
- **Email** өңделмейді, тек оқуға арналған.
- Жоғарғы оң жақтағы «logout» белгішесін басып, жүйеден шығып, /logged_out экраніна оралуға болады.

2.5 Параметрлер (Settings Screen)

- Төменгі навигациялық панельде **Settings** белгішесі.
- **Language** (Тіл) бөлімінде:
  - English
  - Қазақша
  - Русский
- Тіл ауыстырылғаннан кейін, қосымша интерфейсі сол тілдегі мәтіндермен бірден жаңартылады (мысалы, **Log in → Кіру**, **Create Test → Тест құру** және т.б.).
- **Theme** (Тақырып) бөлімінде:
  - Light (Жарық тақырып)
  - Dark (Қараңғы тақырып)
  - Dracula (Арнайы ассортимент: көк-қызғылт түсті схема)
- Таңдаған тақырыбыңыз қосымшаның жалпы көрінісін өзгертеді (фон түсі, түймешік бояуы және т.б.).

3. Мұғалім үшін нұсқаулық

3.1 Канал жасау (Create Channel)

1. **Home Screen** бетіндегі **Create Channel** батырмасын басыңыз.

2. **Channel Name** (Арна атауы) және **Description** (Сипаттама) өрістерін толтырыңыз.
3. **Save** батырмасын басыңыз. Канал Firestore-ға жазылып, сіз автоматты түрде арна экранына (Channel Screen) өтесіз.
4. **Channel Screen → Actions (⋮) → Add Participant** арқылы басқа студенттің email-ін теріп, оны арнаға шақырыңыз.

3.2 Post (Жазба) жариялау

1. **Channel Screen → Posts** қойындысы.
2. «Write a post…» өрісіне мәтін жазыңыз.
3. **Send** («→») батырмасын басыңыз.
4. Жазба Firestore-ға сақталып, барлық арна мүшелерінің тізімінде көрінеді.

3.3 Test (Тест) құру

1. **Channel Screen → Tasks** қойындысы → **Create Test** батырмасын басыңыз.
2. **Test Name** (Тест атауы) өрісін толтырыңыз.
3. **Add Question** батырмасын басып, сұрақтар енгізу бетіне өтіңіз.
   - Әр сұрақ үшін:
     - **Enter question** (Сұрақ мәтіні)
     - **Option 1, Option 2, Option 3, Option 4** – төрт жауап нұсқасын енгізіңіз
     - Дұрыс жауапты таңдаңыз (радиокнопка)
4. Тапсырыс бойынша 10 сұраққа дейін қосуға болады.
5. **Save Test** батырмасын басыңыз. Тест Firestore-ға channels/{channelId}/tests коллекциясына жазылады.

3.4 Тест нәтижелерін бақылау

1. **Channel Screen → Tasks** қойындысы.
2. Тест атауын басыңыз → /TestDetailScreen беті ашылады.
3. Бұл беттегі **Participants' Results** (Нәтижелер) тізімінде:
   - Студент аты
   - Алған % нәтижесі
4. Егер бірнеше студент тапсырып қойса, барлығы бір-бірден тізім арқылы көрсетіледі.

3.5 Chat (Чат) арқылы байланысу

1. Төменгі навигациядан **Chat** батырмасын басыңыз.
2. Жаңа чат ашқыңыз келсе (+ батырмасы) → студент аты бойынша іздеу өрісіне енгізіп, іздеу нәтижесінде шығып тұрған пайдаланушыны таңдаңыз.
3. /ChatDetailScreen беті ашылады, онда жазба теріп, **Send** батырмасын басып хабарлама жіберіңіз.
4. Студент те дереу жауап жолдап, екі жақты диалог жүргізе алады.

3.6 Профильді өңдеу

1. **Person** белгішесін басыңыз → /PersonScreen беті ашылады.
2. **Username** (пайдаланушы аты), **About Me** (өз туралы) өрістерін өзгертіңіз.
3. **Save Changes** батырмасын басып өзгерісті сақтау.
4. Егер шығып кеткіңіз келсе, жоғарғы оң жақтағы **Logout** белгішесін басыңыз → /LoggedOutScreen бетіне оралғыңыз келеді.

4. Қолдану мүмкіндіктері бойынша кесте

Мүмкіндік	Студент	Мұғалім
Тіркелу / Кіру	✅	✅
Каналға қатысу / жасау	✅	✅
Жазба жариялау (Posts)	✅	✅
Тест тапсыру / құру	✅	✅
Тест құру	❌	✅
Тест нәтижесін көру	✅	✅
Private Chat (жеке чат)	✅	✅
Channels ішіндегі Chat (групптық чат)	✅	✅
Профильді өңдеу	✅	✅
Тіл (Language) және Тақырып (Theme) таңдау	✅	✅

5. Қауіпсіздік және қолдау

- **Барлық деректер** – Firestore (Firebase) дерекқорында сақталады, **ең жоғары шифрлау** деңгейімен қорғалады.
- **Authcheck** (авторизациялық тексеріс) арқылы пайдаланушы сессиялары үнемі бақыланады.
- **Рөлдерге сүйене отырып** (student, teacher, owner) Firestore ережелері (security rules) орындалады:
  - Тек арна мүшелері ғана арнадағы жазбаларды, тесттерді, чат хабарламаларын оқи алады.
  - Тек Channel owner (мұғалім) ғана сол арнаға қатысушыларды қосып/ала алады.
  - Тек әр студент өз тест нәтижесін ғана қоя алады (бір рет қана).
- Алдағы болашақтағы жаңартуларда:
  - Push Notifications (жаңа тест, жаңа чат хабарламалары),
  - Offline Cache (тест сұрақтарын жергілікті сақтау),
  - File Upload/Download (оқу материалдарын жүктеу / көру) модульдері қосылады.

6. Қолданбаны қолданудың үлгі сценарийі

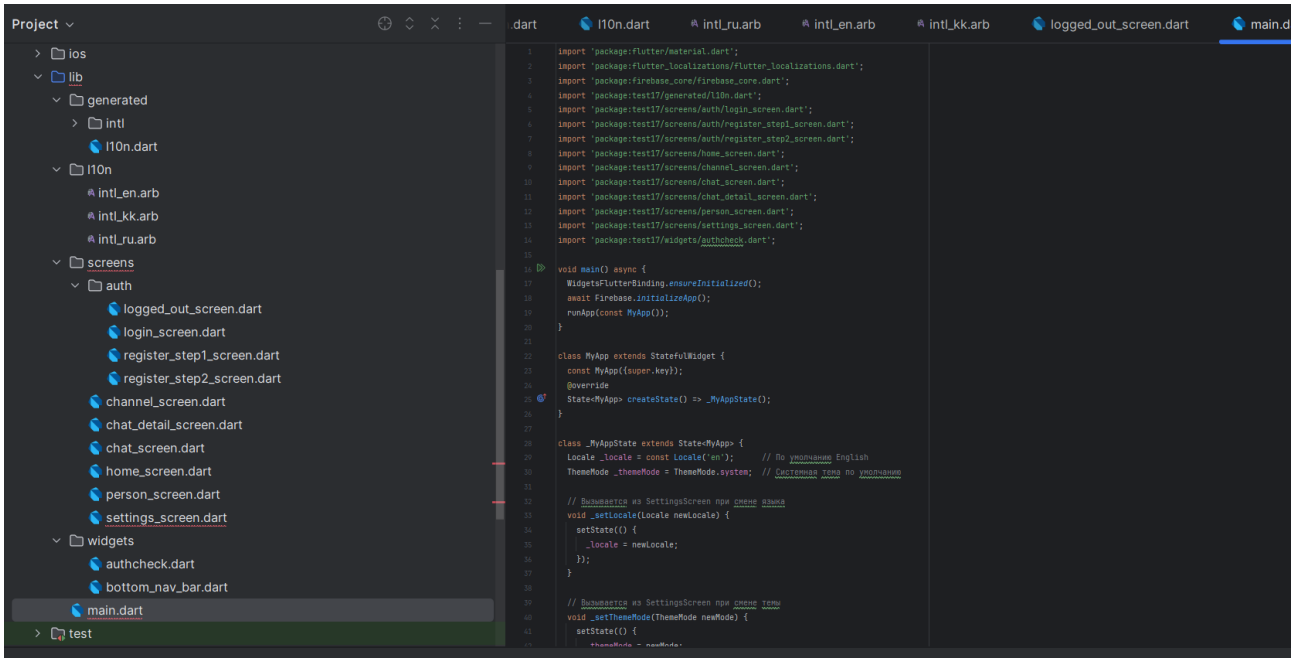
Ә қосымшасының жалғасы

1. **Студент** мобильді қосымшаны ашып, **Register** арқылы νέα тіркеледі:
   - Email/password, Username енгізеді → /register_step2 бетіне өтеді → 4 таңбалы кодты енгізіп /home бетіне аударылады.
2. **Home Screen** бетінде:
   - «Инглиз тілі» деген каналға шақыру коды бар → Join Channel арқылы кодты енгізеді → /channel бетіне өтеді.
3. **Channel Screen** бетінде:
   - **Posts** → «Hello everyone!» жазып, **send** басады → жазба барлық канал мүшелері үшін қолжетімді.
   - **Tasks** → «English Vocabulary Test» тестін таңдайды → сұрақтарға жауап береді → «Submit Test» → нәтижесін **85%** ретінде алады.
4. **Chat Screen** бетіне өтіп, мұғалім атынан сұрақ жібереді: «Келесі тапсырма қашан?»
   - Студент жауап жібереді. Оқытушы дереу жауап жазып, техникалық сұрақтарды талқылайды.
5. **Person Screen** → «Logout» басып, сессиядан шығады → /logged_out экранына оралады.
6. **Login Screen** → Email және парольді енгізіп, қайта кіріседі → Home Screen бетіне оралады.
7. **Settings Screen** → **Language** бөлімінен **Қазақша** тілін таңдап, қосымшаны қазақша интерфейсте пайдалануды жалғастырады.

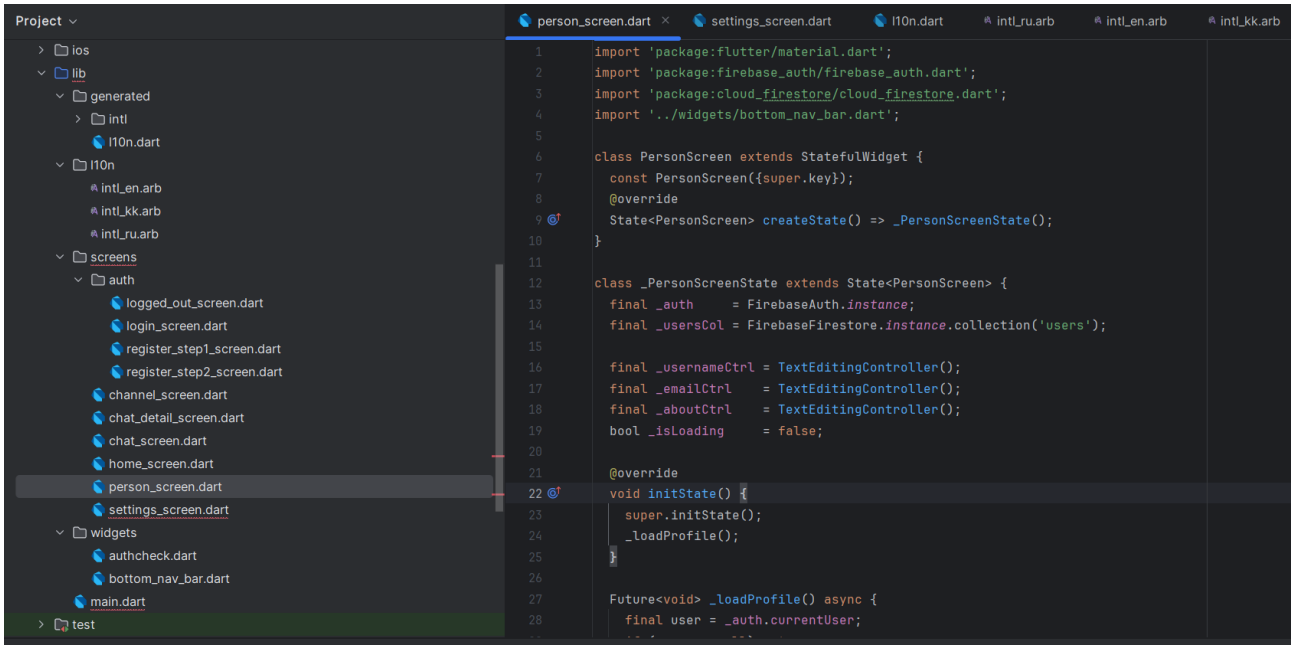
Қосымша Б

Бағдарлама коды

Main



Person Screen



Channel Screen