

АДАПТИВНАЯ МОБИЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ СИСТЕМА: АРХИТЕКТУРА, ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ И МЕХАНИЗМЫ БЕЗОПАСНОСТИ

БИГАЛИЕВА А.З. 

Бигалиева Альфия Замировна – PhD, и.о. доцента, Карагандинский технический университет имени Абылкаса Сагинова, г. Караганда, Казахстан

E-mail: bigalievaalfija@gmail.com, <https://orcid.org/0000-0002-0136-5402>

Аннотация. В статье представлен расширенный и структурированный подход к разработке адаптивной мобильной образовательной системы, ориентированной на формирование индивидуальных траекторий обучения и повышение эффективности цифровой образовательной среды. Рассматривается полный цикл создания мобильного приложения на платформе Flutter, начиная от проектирования пользовательского интерфейса и организации клиентской логики до настройки серверной инфраструктуры и интеграции с облачными сервисами Firebase, обеспечивающими хранение данных, управление пользователями, обработку результатов тестирования и сбор обучающей аналитики. Архитектура системы базируется на принципах чистой архитектуры и паттерне Model-View-ViewModel, что способствует снижению связанности компонентов, повышению модульности, улучшению тестируемости и дальнейшей масштабируемости проекта. В статье представлены ключевые функциональные модули: регистрация и авторизация пользователей, просмотр учебных материалов, механизм адаптивного тестирования, интеллектуальный анализ ошибок, панель аналитики учебного прогресса, а также чатбот, использующий методы искусственного интеллекта для интерактивного сопровождения обучающихся. Значительное внимание уделено вопросам информационной безопасности: применяются методы шифрования, хэширования паролей, распределения ролей и ведения журналов активности. Разработанная система демонстрирует потенциал современных технологий для создания устойчивой, безопасной и персонализированной платформы мобильного обучения.

Ключевые слова: мобильное обучение, архитектура ПО, Flutter, Firebase, адаптивные системы, безопасность данных, аналитика обучения.

Введение

Рост цифровизации образования обуславливает необходимость разработки мобильных платформ, способных адаптироваться под индивидуальные особенности обучающихся. Для эффективного функционирования такие системы должны сочетать удобный интерфейс, гибкую архитектуру, устойчивую серверную инфраструктуру и надежные механизмы защиты данных.

Цель статьи – представить архитектурную модель адаптивной образовательной системы, реализующей индивидуальные траектории обучения на основе стеков Flutter и Firebase, а также продемонстрировать принципы организации данных, логику взаимодействия компонентов и методы обеспечения безопасности.

Материалы и методы исследования

Архитектура представляет собой совокупность взаимосвязанных модулей, обеспечивающих функционирование мобильного приложения, обработку данных, управление пользователями и адаптацию содержания [1]. Она включает четыре ключевых компонента:

- клиентская часть (Flutter-приложение);
- серверная часть и база данных (Firebase Firestore);
- модуль адаптивного обучения;
- административная панель преподавателя.

Такая структура обеспечивает масштабируемость, независимость модулей и возможность гибкого расширения функциональности.

Для повышения устойчивости системы применяется концепция Clean Architecture, предполагающая стогую декомпозицию на уровни данных, бизнес-логики и представления.

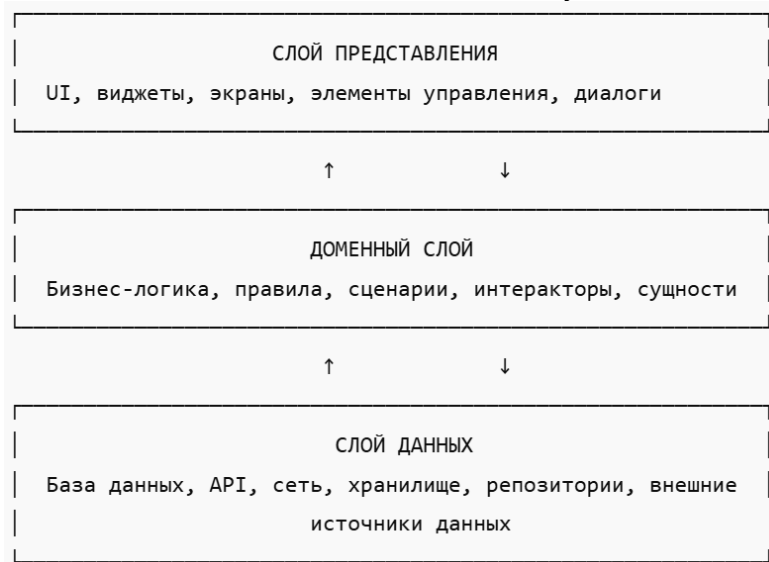


Рисунок 1 –Логическая модель архитектуры приложения (Clean Architecture)

В модели выделены слои:

- Domain (ядро логики),
- Data (данные и репозитории),
- Presentation (виджеты и интерфейс).

Такая организация обеспечивает слабую связанность и удобство сопровождения.

В рамках клиентского приложения применяется архитектурный паттерн Model–View–ViewModel (MVVM). Он позволяет отделить бизнес-логику от пользовательского интерфейса, что повышает тестируемость и модульность.

Flutter, как инструмент кроссплатформенной разработки, выбран благодаря:

- высокой производительности рендеринга;
- декларативному подходу к построению UI;
- поддержке Android и iOS из единого кода;
- нативной интеграции с Firebase.

Это обеспечивает единообразие пользовательского опыта и упрощает разработку [2].

Результаты и их обсуждение

Приложение использует концепцию ORM для описания данных объектами и абстракции запросов к базе. Подход устраняет повторяющийся код и повышает согласованность данных. Фрагмент кода метода добавления объекта в базу данных с использованием Hibernate (DAO-класс)

```
public void add(Type value) throws Exception {
    Session session;
    try {
        session = HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();
        session.save(value);
        session.getTransaction().commit();
    } catch (Exception e) {
        throw e;
    }
}
```

DAO – слой обеспечивает централизованное управление операциями чтения, записи и обновления данных.

Firebase, как облачная инфраструктура, предоставляет [3]:

- Firestore – хранение уроков, вопросов, результатов тестов;
- Authentication – регистрацию и идентификацию;
- Analytics – сбор данных об обучении;
- Cloud Messaging – уведомления;
- хостинг для админ-панели.

Таблица 1 – Основные стабильные плагины Firebase для Flutter

Название плагина	Версия	Firebase продукт	Документация	Исходный код	Поддержка платформ
Authentication	3.3.10	Аутентификация	Есть	firebase_auth	Мобильные ✓, Web ✓, MacOS ✓
Cloud Firestore	3.3.3	База данных	Есть	cloud_firestore	Мобильные ✓, Web ✓, MacOS ✓
Cloud Functions	4.1.1	Облачные функции	Есть	cloud_functions	Мобильные ✓, Web ✓, MacOS ✓
Cloud Messaging	10.0.6	Сообщения	Есть	firebase_messaging	Мобильные ✓, Web ✓, MacOS ✓
Cloud Storage	11.1.3	Хранилище	Есть	firebase_storage	Мобильные ✓, Web ✓, MacOS ✓
Core	3.3.8	Основной пакет	Есть	firebase_core	Мобильные ✓, Web ✓, MacOS ✓
Crashlytics	3.0.0	Аналитика	Есть	firebase_crashlytics	Мобильные ✓, MacOS ✓, Web – (нет поддержки)

Использование пакетов FlutterFire осуществляется через файл pubspec.yaml. Фрагмент конфигурации зависимостей и импортов в Flutter проекте.

dependencies:

flutter:

 sdk: flutter

 http: ^0.13.3

 json_serializable: ^4.1.3

 bloc: ^7.0.0

 flutter_bloc: ^7.0.0

 audioplayers: ^0.18.3

 audio_service: ^0.17.0

 get_it: ^6.0.0

 flutter_svg: ^0.21.0-nullsafety.0

 youtube_player_flutter: ^8.0.0

 firebase_auth: ^1.2.0

 cloud_firestore: ^2.2.0

 provider: ^5.0.0

 flutters_toast: ^8.0.7

 firebase_core: ^1.2.0

import 'package:audio_service/audio_service.dart';

import 'package:audioplayers/audioplayers.dart';

После подключения пакет импортируется в проект, обеспечивая доступ к функционалу Firebase [4].

Для управления пользователями применяется firebase_auth.



Рисунок 2 – Пример использования пакета

Инициализация Firebase в приложении выполняется через `Firebase.initializeApp()`.



Рисунок 3 – Способ инициализации

Таблица 2 – Компонентная структура системы

Модели данных	Представления (View)	Бизнес-логика	Общие сервисы
UserModel информация о пользователе, роль, прогресс	SignInView;	AuthPresenter	DataManager – управление запросами к Firestore
LessonModel структура уроков, вопросы, подсказки	SignUpView	LessonPresenter	AdaptiveLearningEngine – анализ ошибок и подбор сложности
ResultModel – статистика выполнения тестов	LessonListView	QuizPresenter	AIChatService – интеграция с API ИИ
ChatMessageModel – взаимодействие с чатботом	QuizView	AnalyticsPresenter	
	DashboardView	ChatBotPresenter	
	TeacherAnalyticsView		
	ChatBotView		
	AccountView		

Все данные передаются по защищенному протоколу HTTPS с использованием SSL [5].

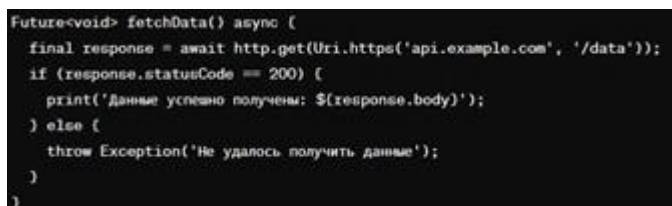


Рисунок 4 – Шифрование данных

Для хранения паролей применяется алгоритм хэширования bcrypt.

```
void registerUser(String username, String password) {  
    final hashedPassword = Bcrypt.hashpw(password, Bcrypt.salt());  
    // Сохранение имени пользователя и хэшированного пароля в базе данных  
}
```

Рисунок 5 – Хэширование паролей

Система распределяет права пользователей по ролям.

```
Future<void> signIn(String email, String password) async {  
    try {  
        await FirebaseAuth.instance.signInWithEmailAndPassword(  
            email: email,  
            password: password,  
        );  
        print('Пользователь успешно вошел в систему');  
    } catch (e) {  
        print('Ошибка аутентификации: $e');  
    }  
}
```

Рисунок 6 – Аутентификация и авторизация

Ведение логов повышает надежность приложения [6].

```
// Пример журналирования событий в Flutter  
void logEvent(String event) {  
    print('Событие: $event');  
    // Запись события в журнал  
}
```

Рисунок 7 – Обработка ошибок и ведение журнала

Заключение

Разработанная архитектура адаптивной образовательной мобильной платформы демонстрирует эффективность интеграции Flutter и Firebase для создания современных интеллектуальных систем обучения. Использование модульности, ORM-подхода, аналитики и безопасных механизмов аутентификации обеспечивает высокую надежность и масштабируемость. Реализованный адаптивный модуль формирует индивидуальные траектории обучения, повышая вовлеченность и успешность пользователей.

Платформа может служить прототипом для разработки новых поколений цифровых образовательных систем, ориентированных на персонализацию и безопасность.

Список литературы

1. Лебедев Б. А. Мобильная разработка на Flutter Dark. СПб.: БХВ- Петербург. – 2018. – 360 с.
2. Сидоров Г. Н. Разработка мобильных приложений. СПб.: Питер. – 2019. – 432 с.
3. Семенчук В. Мобильное приложение как инструмент бизнеса. М: Альпина Диджитал. – 2017. – 270 с.
4. Черняк А. В. Мобильная разработка на платформе Android. М. ДМК Пресс. – 2017. – 352 с.

с.

5. Иванов Д. К., Смирнов А. П. Мобильная разработка: Xamarin, iOS, Android M.: Техносфера. – 2020 – 288 с.

6. Лонг М. Flutter для мобильных разработчиков. Рецепты для реактивного Flutter. СПб.: Питер. – 2020. – 256 с.

References

1. Lebedev B. A. Mobil'naya razrabotka na Flutter Dark. SPb.: BHV- Peterburg. – 2018. – 360 s.

2. Sidorov G. N. Razrabotka mobil'nyh prilozhenij. SPb.: Piter. – 2019. – 432 s.

3. Semenchuk V. Mobil'noe prilozhenie kak instrument biznesa. M: Al'pina Didzhital. – 2017. – 270 s.

4. Chernyak A. V. Mobil'naya razrabotka na platforme Android. M. Dmk Press. –2017. – 352 с.

5. Ivanov D. K., Smirnov A. P. Mobil'naya razrabotka: Hamarin, iOS, Android M.: Tekhnosfera. – 2020 – 288 s.

6. Long M. Flutter dlya mobil'nyh razrabotchikov. Recepty dlya reaktivnogo Flutter. SPb.: Piter. – 2020. – 256 s.

АДАПТИВТІ МОБИЛЬДІ БІЛІМ БЕРУ ЖҮЙЕСІ: СӘУЛЕТ, МӘЛІМЕТТЕР БАЗАСЫН ЖОБАЛАУ ЖӘНЕ ҚАУІПСІЗДІК МЕХАНИЗМДЕРІ

БИГАЛИЕВА А.З. 

Бигалиева Альфия Замировна – PhD, доцент м.а., Әбілқас Сағынов атындағы Қарағанды техникалық университеті, Қарағанды қ., Қазақстан

E-mail: bigalievaaalfija@gmail.com, <https://orcid.org/0000-0002-0136-5402>

Аңдатпа. Мақалада оқытудың жеке траекторияларын қалыптастыруға және цифрлық білім беру ортасының тиімділігін арттыруға бағытталған бейімделген мобильді білім беру жүйесін әзірлеуге кеңейтілген және құрылымдық тәсіл ұсынылған. Пайдаланушы интерфейсін жобалаудан және клиенттік логиканы ұйымдастырудан бастап серверлік инфрақұрылымды орнатуға және деректерді сақтауды, пайдаланушыларды басқаруды, тестілеу нәтижелерін өңдеуді және оқу аналитикасын жинауды қамтамасыз ететін Firebase бұлттық қызметтерімен біріктіруге дейінгі Flutter платформасында мобильді қосымшаны құрудың толық циклі қарастырылады. Жүйенің архитектурасы таза архитектура принциптеріне және компоненттердің байланысын төмендетуге, модульділікті арттыруға, тестілеуді жақсартуға және жобаның одан әрі масштабталуына ықпал ететін Model-View-ViewModel үлгісіне негізделген. Мақалада негізгі функционалды модульдер келтірілген: пайдаланушыларды тіркеу және авторизациялау, оқу материалдарын қарау, бейімделген тестілеу механизмі, қателерді интеллектуалды талдау, Оқу барысын талдау панелі, сондай-ақ оқушыларды интерактивті сүйемелдеу үшін жасанды интеллект әдістерін қолданатын чатбот. Ақпараттық қауіпсіздік мәселелеріне көп көңіл бөлінеді: шифрлау, парольдерді хэштеу, рөлдерді бөлу және белсенділік журналдарын жүргізу әдістері қолданылады. Өзірленген жүйе тұрақты, қауіпсіз және жекелендірілген мобильді оқыту платформасын құру үшін заманауи технологиялардың әлеуетін көрсетеді.

Түйін сөздер: мобильді оқыту, бағдарламалық жасақтама архитектурасы, Flutter, Firebase, адаптивті жүйелер, деректер қауіпсіздігі, оқыту аналитикасы.

ADAPTIVE MOBILE EDUCATIONAL SYSTEM: ARCHITECTURE, DATABASE DESIGN AND SECURITY MECHANISMS

BIGALIYEVA A.Z. 

Bigaliev Alfiya Zamirovna – PhD, acting associate professor, Abylqas Saginov Karaganda technical university, Karaganda, Kazakhstan

E-mail: bigalievaaalfija@gmail.com, <https://orcid.org/0000-0002-0136-5402>

Abstract. The article presents an expanded and structured approach to the development of an adaptive mobile

educational system focused on the formation of individual learning trajectories and improving the effectiveness of the digital educational environment. The full cycle of creating a mobile application on the Flutter platform is considered, starting from designing the user interface and organizing client logic to configuring the server infrastructure and integrating with Firebase cloud services that provide data storage, user management, processing test results and collecting training analytics. The architecture of the system is based on the principles of pure architecture and the Model-View-ViewModel pattern, which helps to reduce component connectivity, increase modularity, improve testability and further scalability of the project. The article presents key functional modules: user registration and authorization, viewing educational materials, adaptive testing mechanism, intelligent error analysis, learning progress analysis panel, as well as a chatbot using artificial intelligence methods for interactive support of students. Considerable attention is paid to information security issues: encryption methods, password hashing, role allocation, and activity logging are used. The developed system demonstrates the potential of modern technologies to create a sustainable, secure and personalized mobile learning platform.

Key words: mobile learning, software architecture, Flutter, Firebase, adaptive systems, data security, learning analytics.