Документация для проекта "Color Match"

Оглавление

**1. Документация по архитектуре**

- Общая структура проекта

- Используемые технологии

- Диаграммы

**2. Документация по коду**

- Комментарии в коде

- README файлы

**3. Руководство пользователя**

- Инструкции по установке

- Руководство по использованию

**4. Руководство разработчика**

- Стартовое руководство

- Стили кодирования

- Инструкции по тестированию

**5. API документация**

- Описание эндпоинтов

- Примеры запросов и ответов

**1. Документация по архитектуре**

Общая структура проекта

- *lib/*: Содержит основные файлы кода Flutter.

- *screens/:* Содержит экраны приложения (*start\_screen.dart, login\_screen.dart, game\_screen.dart, leaderboard\_screen.dart*).

- *firebase\_options.dart*: Содержит параметры конфигурации Firebase.

- *main.dart*: Главный файл приложения.

- *android/* *и* *ios/:* Платформоспецифические файлы для Android и iOS.

- *pubspec.yaml*: Файл конфигурации для Flutter проекта.

Используемые технологии

- *Flutter/Dart*: Для разработки кроссплатформенного мобильного приложения.

- *Firebase*: Для аутентификации пользователей и хранения данных.

- *Firebase Authentication*: Для регистрации и авторизации пользователей.

- *Cloud Firestore*: Для хранения данных, таких как лучшие результаты игроков.

- *Firebase CLI*: Для настройки и развертывания Firebase.

Диаграммы

Диаграмма архитектуры приложения

+-------------------------+  
| Flutter App |  
|-------------------------|  
| |  
| +---------------------+ |  
| | UI Layer | |  
| | +-----------------+ | |  
| | | Screens | | |  
| | | (Start, Login, | | |  
| | | Game, Leadrbrd)| | |  
| | +-----------------+ | |  
| +---------------------+ |  
| |  
| +---------------------+ |  
| | Business Logic | |  
| | (Dart classes) | |  
| +---------------------+ |  
| |  
+-------------------------+  
 |  
 v  
+-------------------------+  
| Firebase |  
|-------------------------|  
| |  
| +---------------------+ |  
| | Authentication | |  
| +---------------------+ |  
| |  
| +---------------------+ |  
| | Firestore DB | |  
| +---------------------+ |  
| |  
+-------------------------+

**2. Документация по коду**

Комментарии в коде

Каждая функция и важный блок кода снабжены комментариями, объясняющими их назначение и логику. Пример:

// Запускает таймер, который обновляет значение каждую 0,1 секунду  
void \_startTimer() {  
 const oneTenth = const Duration(milliseconds: 100);  
 \_timer = Timer.periodic(oneTenth, (timer) {  
 setState(() {  
 if (\_timerValue <= 0) {  
 \_timer?.cancel();  
 \_isGameOver = true;  
 \_animationController?.forward();  
 \_saveScore(); // Сохранение результатов после окончания игры  
 } else {  
 \_timerValue -= 0.1;  
 }  
 });  
 });  
}

README файлы

Основной README файл

# Color Match Game  
  
## Описание  
Color Match Game - это аркадная игра, направленная на улучшение навыков сочетания цветов у пользователей.  
  
## Стек технологий  
- Flutter/Dart  
- Firebase (Authentication, Firestore)  
- ASP.NET Core (опционально)  
  
## Установка и запуск  
1. Клонируйте репозиторий:  
- git clone <repository\_url>  
2. Перейдите в каталог проекта:  
- cd color\_match\_game  
3. Установите зависимости:  
- flutter pub get  
4. Запустите приложение:  
- flutter run

Конфигурация Firebase

- Создайте проект в Firebase и настройте Authentication и Firestore.

- Скачайте файл конфигурации `google-services.json` для Android и `GoogleService-Info.plist` для iOS и добавьте их в соответствующие каталоги.

- Используйте Firebase CLI для инициализации проекта.

Контрибьютинг

- Приветствуются любые предложения и запросы на изменения.

**3. Руководство пользователя**

Инструкции по установке

1. \*\*Клонирование репозитория\*\*  
git clone <repository\_url>  
  
2. Установка зависимостей  
flutter pub get  
  
3. Конфигурация Firebase  
  
Создайте проект в Firebase и настройте Authentication и Firestore.  
Скачайте файл конфигурации google-services.json для Android и GoogleService-Info.plist для iOS и добавьте их в соответствующие каталоги.  
  
4. Запуск приложения  
flutter run

Руководство по использованию

1. Запуск игры

- Откройте приложение и нажмите "Начать" на стартовом экране.

2. Регистрация и вход

- Для регистрации нового пользователя введите email, пароль и никнейм.

- Для входа в существующий аккаунт введите email и пароль.

3. Игровой процесс

- Отвечайте на вопросы о сочетаемости цветов, выбирая "да" или "нет".

- Следите за таймером и количеством очков.

- После окончания игры просмотрите свой результат и при желании начните игру заново.

4. Таблица лидеров

- Нажмите "Таблица лидеров" после окончания игры, чтобы просмотреть результаты других игроков.

**4. Руководство разработчика**

Стартовое руководство

1. Клонирование репозитория

git clone <repository\_url>

2. Установка зависимостей

flutter pub get

3. Настройка Firebase

- Следуйте инструкциям из основного README для настройки Firebase.

4. Запуск приложения

flutter run

Стили кодирования

- Используйте CamelCase для имен переменных и функций.

- Обязательно комментируйте сложные участки кода.

- Соблюдайте отступы и пробелы для улучшения читаемости.

Инструкции по тестированию

1. Запуск тестов

flutter test

2. Написание тестов

- Разрабатывайте тесты для каждого экрана и основной логики приложения.

- Используйте Flutter тестовые утилиты для написания и запуска тестов.

**5. API документация**

Описание эндпоинтов

Firebase Authentication

- Регистрация: `createUserWithEmailAndPassword(email, password)`

- Вход: `signInWithEmailAndPassword(email, password)`

Firestore

- Сохранение результата: `/scores/{userId}`

- Метод: `set`

- Параметры: `bestScore, nickname`

Примеры запросов и ответов

Регистрация пользователя

Запрос:

UserCredential userCredential = await FirebaseAuth.instance.createUserWithEmailAndPassword(  
email: email,  
password: password,  
);

Ответ:

{  
 "user": {  
 "uid": "unique\_user\_id",  
 "email": "user@example.com"  
 }  
}