deployment  
 %% Client Side  
 node "User's Mobile Device" {  
 component[Flutter Mobile App] as FlutterApp {  
 artifact "UI Components" as UI  
 artifact "Business Logic" as BL  
 artifact "Local Storage (Hive)" as LS  
 artifact "Service Layer" as SL  
 }  
 }  
  
 %% Firebase Cloud Services  
 node "Firebase Cloud Platform" {  
 component[Firebase Authentication] as Auth  
 component[Firebase Firestore] as Firestore {  
 artifact "User Data" as UserData  
 artifact "Mood Entries" as MoodData  
 artifact "User Settings" as SettingsData  
 }  
 component[Firebase Storage] as Storage {  
 artifact "User Media" as MediaData  
 }  
 component[Firebase Cloud Functions] as CloudFunctions {  
 artifact "Data Sync" as SyncFunc  
 artifact "Analytics Processing" as AnalyticsFunc  
 artifact "Notifications" as NotificationFunc  
 }  
 component[Firebase Cloud Messaging] as FCM  
 }  
  
 %% External APIs  
 node "Third-Party Services" {  
 component[Weather API Service] as WeatherAPI  
 component[Geolocation Service] as GeoAPI  
 component[Analytics Services] as ThirdPartyAnalytics  
 }  
  
 %% Connections/Communications  
 UI -- BL  
 BL -- SL  
 BL -- LS  
   
 SL -- Auth : HTTPS  
 SL -- Firestore : HTTPS  
 SL -- Storage : HTTPS  
 SL -- FCM : HTTPS  
 SL -- WeatherAPI : HTTPS  
 SL -- GeoAPI : HTTPS  
   
 Auth -- Firestore : Internal  
 Firestore -- CloudFunctions : Internal  
 CloudFunctions -- FCM : Internal  
 CloudFunctions -- ThirdPartyAnalytics : HTTPS

## Figure 4.8: Deployment Diagram - Pro Mood Tracker Application

This deployment diagram illustrates how the Pro Mood Tracker application components are distributed across physical infrastructure, showing the runtime architecture.

### Client Side

**User’s Mobile Device:** - **Flutter Mobile App**: The mobile application installed on the user’s device - **UI Components**: All user interface elements - **Business Logic**: Application logic and state management - **Local Storage (Hive)**: NoSQL database for offline data persistence - **Service Layer**: Interfaces with external services and APIs

### Cloud Infrastructure

**Firebase Cloud Platform:** - **Firebase Authentication**: Manages user identity and access control - **Firebase Firestore**: NoSQL document database storing: - User profiles and authentication data - Mood entries with associated metadata - User preferences and application settings - **Firebase Storage**: Object storage for user-generated media - **Firebase Cloud Functions**: Serverless functions handling: - Data synchronization between devices - Background analytics processing - Push notification management - **Firebase Cloud Messaging**: Push notification delivery service

### External Services

**Third-Party Services:** - **Weather API Service**: Provides current and forecast weather data - **Geolocation Service**: Provides location lookup and reverse geocoding - **Analytics Services**: External services for advanced data processing

### Communication Patterns

1. The mobile app communicates with Firebase services via secure HTTPS connections
2. Firebase services communicate internally through the Firebase platform
3. Cloud Functions interact with external APIs for enhanced functionality
4. The mobile app stores data locally for offline operation, syncing when connectivity is restored
5. Push notifications are delivered via Firebase Cloud Messaging

This deployment architecture provides: - Scalable cloud infrastructure that grows with user demand - Robust offline capabilities through local storage - Cross-device synchronization via cloud services - Secure authentication and data storage - Serverless backend requiring minimal maintenance