classDiagram  
 class User {  
 -String id  
 -String username  
 -String email  
 -String passwordHash  
 -Date createdAt  
 -Date lastLogin  
 -Map preferences  
 -int streakDays  
 -int totalPoints  
 +login(email, password): boolean  
 +register(username, email, password): User  
 +updateProfile(profileData): boolean  
 +getStats(): UserStats  
 +resetPassword(email): boolean  
 }  
  
 class MoodEntry {  
 -String id  
 -String userId  
 -Date entryDate  
 -Time entryTime  
 -int moodRating  
 -String notes  
 -Map locationData  
 -Map weatherData  
 -List~Activity~ activities  
 +create(userId, moodData): MoodEntry  
 +update(moodData): boolean  
 +delete(): boolean  
 +addActivity(activityId, impactRating): boolean  
 +removeActivity(activityId): boolean  
 }  
  
 class Activity {  
 -String id  
 -String name  
 -String icon  
 -String category  
 +create(activityData): Activity  
 +update(activityData): boolean  
 +delete(): boolean  
 +getByCategory(category): List~Activity~  
 }  
  
 class Achievement {  
 -String id  
 -String name  
 -String description  
 -int pointsValue  
 -String badgeIcon  
 -String criteriaType  
 -Map criteriaValue  
 +checkEligibility(user): boolean  
 +award(userId): boolean  
 }  
  
 class UserAchievement {  
 -String userId  
 -String achievementId  
 -Date earnedDate  
 -boolean isViewed  
 +markViewed(): boolean  
 +getAllForUser(userId): List~UserAchievement~  
 }  
  
 class Theme {  
 -String id  
 -String name  
 -String description  
 -Map colorScheme  
 -boolean isSystem  
 +apply(userId): boolean  
 +create(themeData): Theme  
 +update(themeData): boolean  
 +delete(): boolean  
 }  
  
 class Reminder {  
 -String id  
 -String userId  
 -Time reminderTime  
 -List~String~ activeDays  
 -boolean isEnabled  
 -String message  
 +enable(): boolean  
 +disable(): boolean  
 +update(reminderData): boolean  
 +delete(): boolean  
 +checkAndNotify(): void  
 }  
  
 class AnalyticsService {  
 +generateMoodInsights(userId, timeframe): MoodInsights  
 +calculateTrends(userId, timeframe): TrendData  
 +identifyPatterns(userId): List~Pattern~  
 +exportData(userId, format): File  
 }  
  
 class NotificationService {  
 +sendReminder(userId, message): boolean  
 +notifyAchievement(userId, achievementId): boolean  
 +scheduleReminders(): void  
 +cancelReminder(reminderId): boolean  
 }  
  
 class AuthService {  
 +authenticate(email, password): AuthToken  
 +validateToken(token): boolean  
 +refreshToken(token): AuthToken  
 +revokeToken(token): boolean  
 }  
  
 User "1" -- "many" MoodEntry : records  
 User "1" -- "many" Reminder : sets  
 User "1" -- "many" UserAchievement : earns  
 MoodEntry "many" -- "many" Activity : includes  
 Achievement "1" -- "many" UserAchievement : awarded as  
 User -- AuthService : authenticates through  
 User -- AnalyticsService : analyzes data with  
 User -- NotificationService : receives alerts from  
 Reminder -- NotificationService : triggered through

## Figure 4.3: Class Diagram

The class diagram illustrates the structure of the Pro Mood Tracker application, showcasing the primary classes, their attributes, methods, and relationships that form the application’s architecture.

### Core Domain Classes

1. **User**
   * Central class representing application users
   * Contains authentication and profile information
   * Tracks engagement metrics and preferences
   * Provides methods for account management
2. **MoodEntry**
   * Represents individual mood records
   * Stores contextual data (location, weather, time)
   * Contains the core mood rating and notes
   * Manages relationships with activities
3. **Activity**
   * Represents categorized activities that influence mood
   * Provides organization through categories
   * Includes visual representation through icons
4. **Achievement**
   * Implements gamification elements
   * Defines criteria for earning achievements
   * Awards points and badges to users

### Supporting Classes

1. **Theme**
   * Enables UI customization
   * Stores color schemes for application appearance
   * Distinguishes between system and user-created themes
2. **Reminder**
   * Manages notification scheduling
   * Supports recurring reminders on specified days
   * Contains customizable messages

### Junction Classes

1. **UserAchievement**
   * Connects users with their earned achievements
   * Tracks achievement status and earning date
   * Provides notification management for new achievements

### Service Classes

1. **AnalyticsService**
   * Processes mood data to generate insights
   * Identifies patterns and calculates trends
   * Exports data for external analysis
2. **NotificationService**
   * Handles all application notifications
   * Manages reminder scheduling and delivery
   * Sends achievement notifications
3. **AuthService**
   * Manages user authentication and security
   * Handles token generation and validation
   * Provides secure access to the application

### Key Relationships

* Each user can record many mood entries and set multiple reminders
* Mood entries can include multiple activities
* Users earn achievements based on their engagement and mood tracking behavior
* Service classes provide cross-cutting functionality to the domain classes

This class structure supports both the core functionality of mood tracking and the extended features like achievements, analytics, and customization that enhance the user experience.