# Table 3.4: Data Storage Structure and Organization

| Storage Key | Data Structure | Size Estimation | Update Frequency | Optimization Techniques |
| --- | --- | --- | --- | --- |
| proMoodTracker:user | json { "id": "uuid-string", "username": "string", "email": "string", "createdAt": "ISO-date-string", "points": number, "settings": { "theme": "theme-name", "enableWeatherTracking": boolean, "enablePredictions": boolean, "location": { "city": "string", "country": "string", "lat": number, "lon": number } } } | 1-5 KB per user | Low (Profile updates) | - Minimal nested structure- Separate user settings from core data- Only essential fields stored |
| proMoodTracker:moodEntries | json [ { "id": "uuid-string", "date": "ISO-date-string", "timeOfDay": "period-string", "mood": "mood-string", "notes": "string", "customCategory": "string", "time": "HH:MM", "journal": "string", "activities": ["string"], "weather": { "temperature": number, "condition": "string", "humidity": number } } ] | 0.5-2 KB per entry(~500 KB for 1 year daily) | High (Daily entries) | - Optional fields omitted when empty- Pagination for large histories- Archive older entries separately- Compress journal entries |
| proMoodTracker:badges | json [ { "id": "uuid-string", "name": "string", "description": "string", "imageUrl": "string", "criteria": "string", "points": number, "tier": "string", "earnedAt": "ISO-date-string" } ] | 0.5-1 KB per badge(~25 KB for all badges) | Medium (Achievement unlocks) | - Store only earned badges- Reference badges by ID when possible- Lazy load badge images |
| proMoodTracker:themes | json [ { "id": "uuid-string", "name": "string", "primary": "hex-string", "secondary": "hex-string", "background": "hex-string", "paper": "hex-string", "text": "hex-string", "createdAt": "ISO-date-string" } ] | 0.3-0.5 KB per theme(~5 KB for all themes) | Low (Theme creation) | - Store only custom themes- Built-in themes defined in code- Minimal color definitions |
| proMoodTracker:stats | json { "calculatedAt": "ISO-date-string", "averageMoodByDay": { "Monday": number, "Tuesday": number, ... }, "averageMoodByTimeOfDay": { "morning": number, "afternoon": number, ... }, "moodCountByType": { "Very Good": number, "Good": number, ... }, "mostFrequentMood": "string", "moodTrend": "string", "moodCorrelations": { "weather": { ... }, "dayOfWeek": { ... } } } | 2-10 KB | Medium (Recalculated on new entries) | - Cache calculated statistics- Recalculate only on significant changes- Store only significant correlations |
| proMoodTracker:streaks | json { "currentDaily": number, "longestDaily": number, "startDate": "ISO-date-string", "lastEntryDate": "ISO-date-string", "history": [ { "type": "string", "count": number, "startDate": "ISO-date-string", "endDate": "ISO-date-string" } ] } | 0.5-2 KB | High (Daily updates) | - Store only current and longest streaks- Limited streak history- Calculate streaks on demand when possible |
| proMoodTracker:notifications | json [ { "id": "uuid-string", "type": "string", "message": "string", "read": boolean, "createdAt": "ISO-date-string", "expiresAt": "ISO-date-string", "data": { ... } } ] | 0.2-0.5 KB per notification(~10 KB for active notifications) | Medium (New achievements, reminders) | - Auto-expire old notifications- Limit maximum stored notifications- Clean up read notifications periodically |
| proMoodTracker:activities | json [ { "id": "uuid-string", "name": "string", "icon": "string", "category": "string", "count": number, "lastUsed": "ISO-date-string" } ] | 0.1-0.2 KB per activity(~10 KB for all activities) | Medium (Activity tracking) | - Store only user-defined activities- Built-in activities defined in code- Track usage count for optimization |
| proMoodTracker:session | json { "lastActive": "ISO-date-string", "currentView": "string", "formData": { ... }, "filters": { ... } } | 1-5 KB | High (Session updates) | - Store in sessionStorage instead of localStorage- Minimal essential data only- Clear on session end |
| proMoodTracker:cache | Various cached data | 5-20 KB | High (Dynamic caching) | - TTL-based expiration- LRU cache eviction policy- Size limits per cache category |