

# シンプル日記アプリケーション - 設計図

## 1. 画面遷移図

```
stateDiagram-v2
    [*] --> トップページ

    state "未ログイン" as Guest {
        トップページ --> ログイン
        トップページ --> 新規登録
        ログイン --> トップページ: ログイン成功
        新規登録 --> ログイン: 登録成功
    }

    state "ログインユーザー" as User {
        state "一般ユーザー機能" as Normal {
            トップページ --> ユーザー設定
            ユーザー設定 --> トップページ
        }
    }

    state "管理者" as Admin {
        state "管理者機能" as AdminFunc {
            トップページ --> ユーザー管理
            ユーザー管理 --> トップページ
        }
    }

    トップページ --> [*]: ログアウト
```

## 2. クラス図

```
classDiagram
    class User {
        +int id
        +string userid
        +string name
        +string password
        +bool is_admin
        +bool is_locked
        +int login_attempts
        +datetime last_login_attempt
        +datetime created_at
        +validate_password()
        +check_lock_status()
    }

    class Entry {
```

```
+int id
+int user_id
+string title
+string content
+datetime created_at
+datetime updated_at
+create()
+update()
+delete()
}

class UserManager {
    +lock_user()
    +unlock_user()
    +toggle_admin()
    +update_login_attempts()
}

User "1" -- "*" Entry : creates
UserManager -- User : manages
```

### 3. シーケンス図 (ログインプロセス)

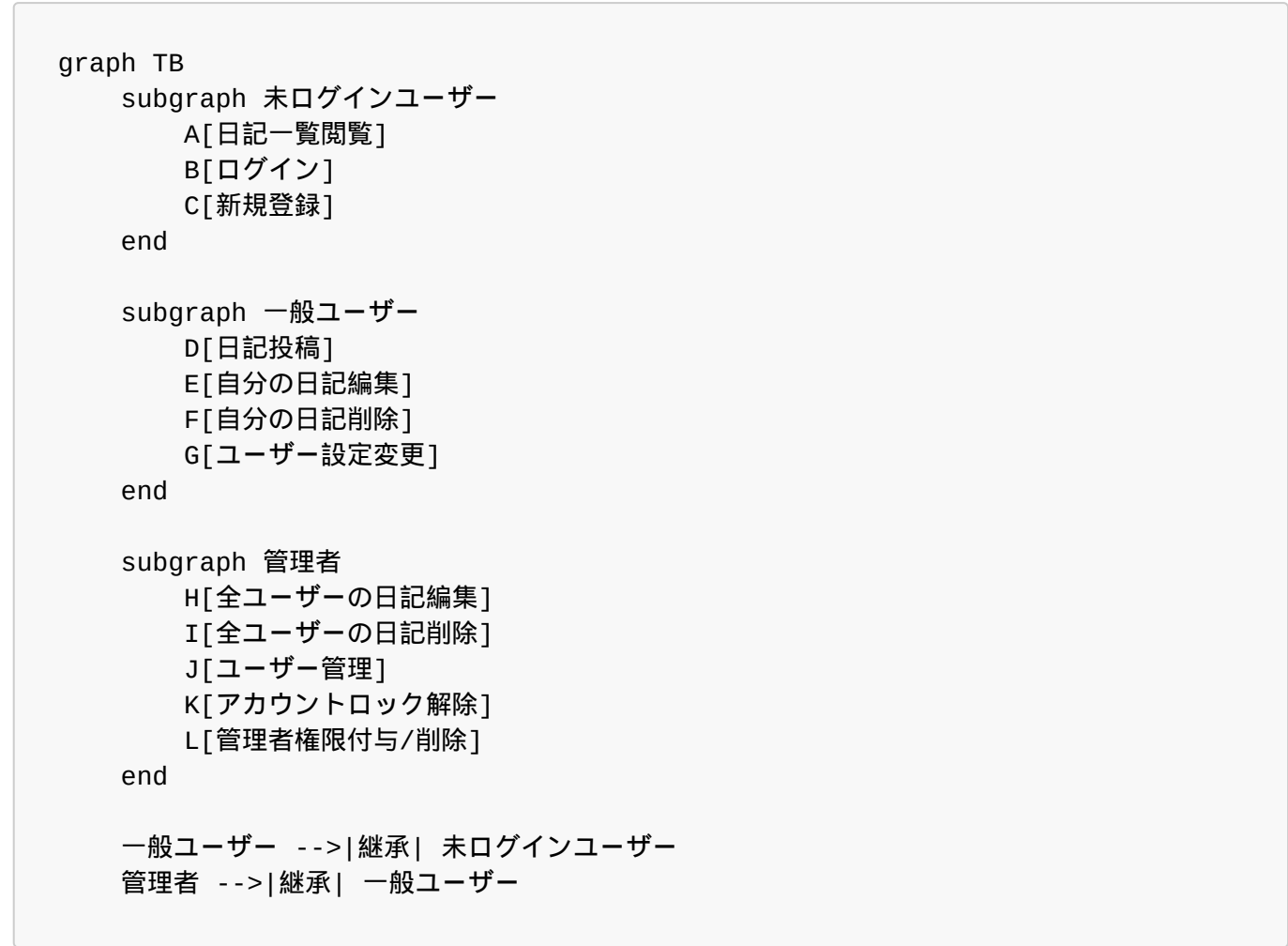
```
sequenceDiagram
    actor User
    participant Frontend
    participant Backend
    participant Database

    User->>Frontend: ログイン情報入力
    Frontend->>Backend: POST /api/login
    Backend->>Database: ユーザー検索
    Database-->>Backend: ユーザー情報

    alt ユーザーが存在する
        alt パスワードが正しい
            Backend->>Database: ログイン試行回数リセット
            Backend-->>Frontend: 成功レスポンス
            Frontend->>Frontend: トップページへリダイレクト
        else パスワードが間違っている
            Backend->>Database: ログイン試行回数増加
            alt 試行回数 >= 3
                Backend->>Database: アカウントをロック
                Backend-->>Frontend: ロックエラー
            else 試行回数 < 3
                Backend-->>Frontend: 認証エラー
            end
        end
    end

    else ユーザーが存在しない
        Backend-->>Frontend: 認証エラー
    end
end
```

## 4. ユースケース図



## 5. ERD (Entity Relationship Diagram)



```
        datetime updated_at "更新日時"
    }

    users ||--o{ entries : "creates"
```

6. アクティビティ図

ログインプロセス

```
flowchart TD
    A[開始] --> B[ログインフォーム表示]
    B --> C[ユーザーID/パスワード入力]
    C --> D{ユーザーが存在する?}
    D -->|Yes| E{アカウントがロックされている?}
    D -->|No| F[エラー: ユーザーが見つかりません]
    F --> B
    E -->|Yes| G[エラー: アカウントがロックされています]
    G --> B
    E -->|No| H{パスワードが一致する?}
    H -->|Yes| I[ログイン試行回数をリセット]
    I --> J[セッション作成]
    J --> K[トップページへリダイレクト]
    K --> L[終了]
    H -->|No| M[ログイン試行回数を増加]
    M --> N{試行回数 >= 3?}
    N -->|Yes| O[アカウントをロック]
    O --> P[エラー: アカウントがロックされました]
    P --> B
    N -->|No| Q[エラー: パスワードが正しくありません]
    Q --> B
```

7. コンポーネント図

システムアーキテクチャ

```
graph TB
    subgraph フロントエンド
        A[main.css]
        B[admin.css]
        C[script.js]
    end

    subgraph テンプレート
        D[index.html]
        E[login.html]
        F[register.html]
    end
```

```
    G[settings.html]
    H[admin.html]
end

subgraph バックエンド
    I[app.py]
    J[schema.sql]
end

subgraph データベース
    K[diary.db]
end

%% フロントエンド依存関係
D --> A
E --> A
F --> A
G --> A
H --> A
H --> B

%% テンプレート依存関係
D --> C
E --> C
F --> C
G --> C
H --> C

%% バックエンド依存関係
I --> J
I --> K

style A fill:#f9f,stroke:#333,stroke-width:2px
style B fill:#f9f,stroke:#333,stroke-width:2px
style C fill:#bbf,stroke:#333,stroke-width:2px
style I fill:#bfb,stroke:#333,stroke-width:2px
style K fill:#ff9,stroke:#333,stroke-width:2px
```

## ディレクトリ構造

```
/
├─ app.py           # メインアプリケーション
├─ schema.sql       # データベーススキーマ
├─ diary.db         # SQLiteデータベース
├─ requirements.txt # 依存パッケージリスト
├─ static/          # 静的ファイル
│   ├─ main.css     # 共通スタイル
│   ├─ admin.css    # 管理画面スタイル
│   └─ script.js    # クライアントサイドスクリプト
└─ templates/       # HTMLテンプレート
    └─ index.html   # トップページ
```

```
└─ login.html      # ログインページ
└─ register.html   # ユーザー登録ページ
└─ settings.html   # ユーザー設定ページ
└─ admin.html      # 管理者ページ
```

## 8. 補足：データベース制約

### usersテーブル

- **id**: 自動増分の主キー
- **userid**: 一意制約、半角英数字のみ（4-20文字）
- **name**: 文字種自由（3-20文字）
- **password**: 8-20文字（大文字・小文字・数字を含む）
- **is\_admin**: デフォルトfalse
- **is\_locked**: デフォルトfalse
- **login\_attempts**: デフォルト0
- **created\_at**: NOT NULL

### entriesテーブル

- **id**: 自動増分の主キー
- **user\_id**: 外部キー（users.id）、NOT NULL
- **title**: NOT NULL
- **content**: NOT NULL
- **created\_at**: NOT NULL
- **updated\_at**: NULL許容（更新時のみ設定）