



# Laravelブログシステム 開発画面設計書

画面レイアウト・機能要件・アーキテクチャ概要

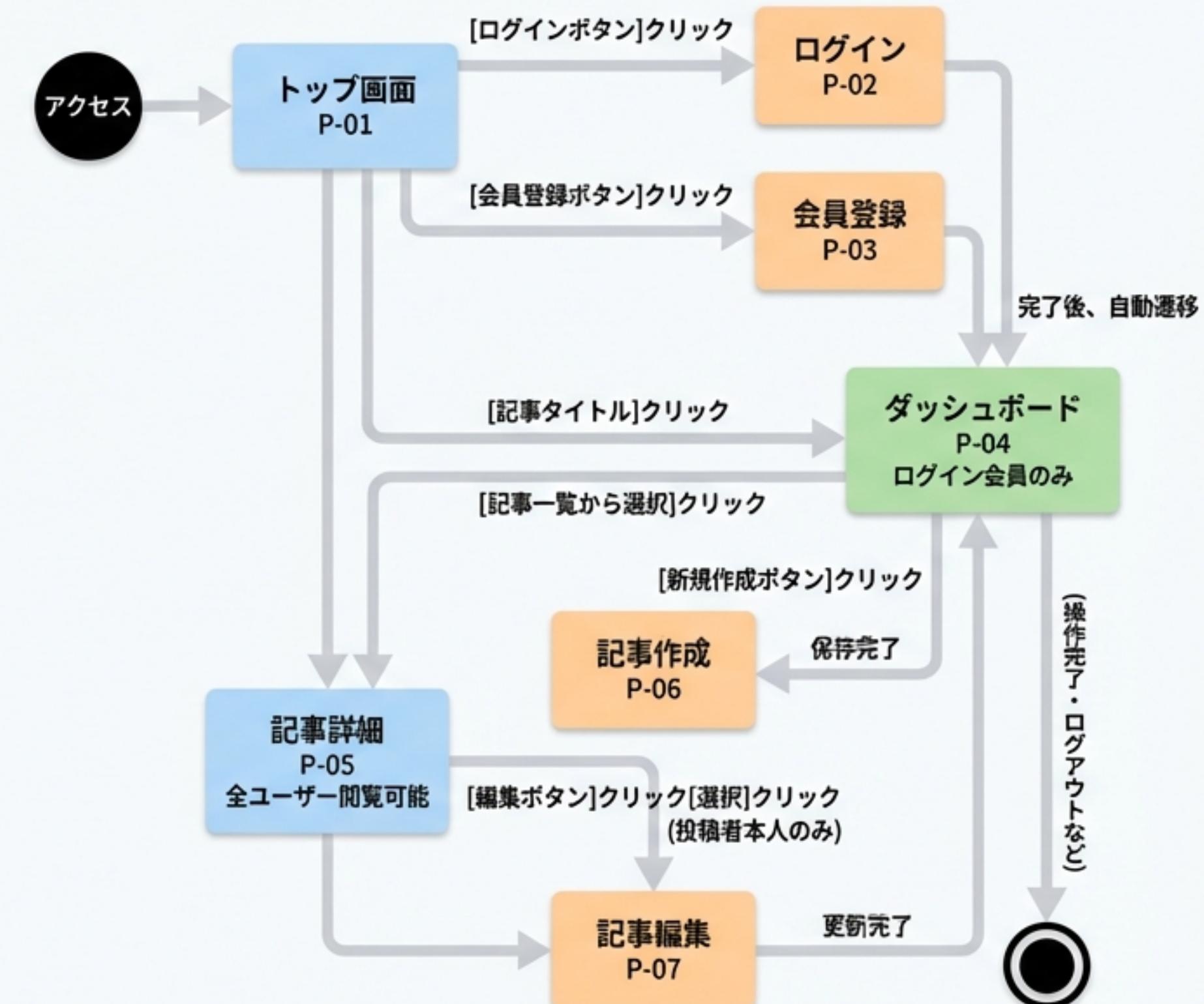
---

**Project Scope:** PHP(Laravel)によるWebブログ機能構築

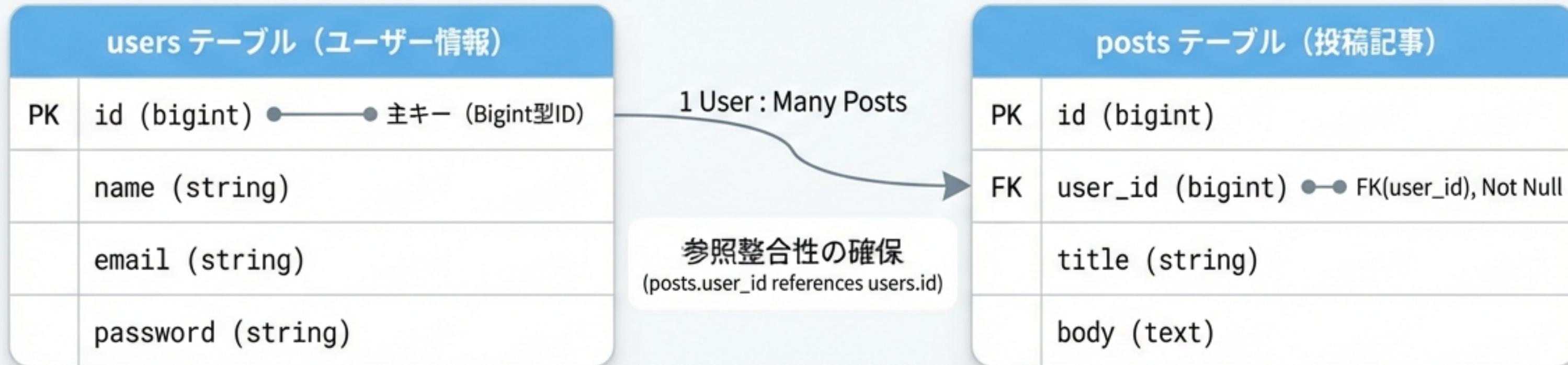
**Document Focus:** UIレイアウト全7画面と実装ロジック

# システム構成と全体フロー

- Backend: PHP 8.x, Laravel 12
- Frontend: Blade, Tailwind CSS
- Database: MySQL 8.0
- Infra/CI: Docker Compose, GitHub Actions



# データベース設計 (ER Diagram & Schema)



# UIデザイン・バリデーション方針

## Design Concept (JIN Theme)



### サンプル記事のタイトル：JINテーマの活用

ここに記事の概要が入ります。クリーンなデザインで情報を整理します。



ベースは白、アクセントにソフトな青/緑を使用

## Validation Rules

お名前

! この項目は必須です

- Immediate Feedback: 各入力項目の直下にエラーメッセージを表示
- Data Retention: エラー時も old() ヘルパで入力値を保持
- Security: CSRFトークン必須、XSS対策

# P-01 トップ画面 (Top Page)

The screenshot shows a wireframe of a top page template. At the top is a header bar with a logo icon and four menu items: Home, About, About, and Contact. Below the header is a grid of six article cards, arranged in two rows of three. Each card has a large gray placeholder image at the top, followed by the title "記事タイトルサンプル", a short excerpt "ここに記事の概要が入ります。クリーンなデザインで情...", and the date "2023.10.20". At the bottom of the page is a navigation bar with page numbers 1, 2, 3, and a right arrow.

**Route:** GET /

**Controller:** PostController@index

**Logic:**

- latest()>paginate(10)
- Str::limit() for excerpts
- @auth blade check for header

# P-02 ログイン (Login)

ログイン

Email Address

Password

ログイン状態を保持

ログイン

アカウントをお持ちでない方はこちら

**Route:** GET/POST /login

**Controller:**

AuthenticatedSessionController

**Validation:**

- Email: Required, DNS, Exists
- Password: Required

**Security:**

session()->regenerate()

# P-03 会員登録 (Registration)

The screenshot shows a web browser window with a registration form. At the top, there is a navigation bar with a logo and links for Home, About, and Contact. The main content area has a title '会員登録' (Registration) and contains four input fields: 'Name', 'Email Address', 'Password', and 'Confirm Password'. Below these fields is a large green button labeled '登録する' (Register). The 'Name' field is currently active, indicated by a blue border.

**Route:** GET/POST /register

**Controller:**

RegisteredUserController

**Validation:**

- Name: Max 255
- Email: Unique:users
- Password: Min 8, Confirmed

**Logic:** Hash::make(), Auto-login

# P-04 ダッシュボード (Dashboard)

The screenshot shows a web application dashboard. At the top, there is a navigation bar with a logo on the left and links for Home, About, Contact, and another About link. Below the navigation bar, the text "Welcome, User-san" is displayed. A blue button labeled "新規投稿を作成" (Create New Post) is visible. Below this, a table lists two posts:

Laravelの勉強について	2023.10.25	[Edit]  [Delete]
Docker環境構築	2023.10.24	[Edit]  [Delete]

**Route:** GET /dashboard

**Middleware:** auth

**Logic:**

- auth()->user()->posts()
- Fetch only own posts
- Delete triggers  
PostController@destroy

# P-06 記事作成 (Create Post)

The screenshot shows a web application window with a title bar containing three dots, a logo icon, and a navigation menu with links for Home, About, Contact, and another About link. Below the menu is a large input field with a light blue border and rounded corners, containing the placeholder text "記事タイトルを入力". At the bottom of the page are two buttons: a blue rounded rectangle labeled "公開する" and a white button labeled "キャンセル".

**Route:** GET/POST /posts/create

**Controller:**

PostController@create/store

**Validation:**

- Title: Required, Max 255
- Body: Required

**Logic:** \$request->user()->posts  
()->create()

# P-05 記事詳細 (Article Detail)

The screenshot shows a web browser window with a header containing a logo, Home, About, Contact, and a search bar. The main content area displays an article titled "Laravel 12の新機能" by "Author: Yamato | 2023.10.26". Below the title are two buttons: "Edit" and "Delete", both enclosed in a dotted red box. To the right of these buttons, the text "Visible only if Auth::id() === post->user\_id" is displayed. The article content is placeholder text in Japanese. At the bottom, there are three more paragraphs of placeholder text.

Laravel 12の新機能

Author: Yamato | 2023.10.26

**Edit** **Delete**

Visible only if  
Auth::id() === post->user\_id

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

**Route:** GET /posts/{id}

**Controller:**

PostController@show

**Logic:**

- findOrFail(\$id)
- Authorization Policy Check

# P-07 記事編集 (Edit Post)

The screenshot shows a web application interface. At the top, there is a header bar with a logo on the left and navigation links: Home, About, About, and Contact. Below the header, the main content area has a title "Laravel 12の新機能" (New features of Laravel 12) in a blue-bordered box. The main body contains placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, arnimin veniam, quis nostrud exercitation ullamco laboris nrat nisi ut aliquip nisi ut commodo conseciaboris nisi ut aliquip ex ea". At the bottom left, there is a blue button labeled "更新する" (Update).

**Route:** PUT/PATCH /posts/{id}

**Controller:**

PostController@edit/update

**Data Binding:**

- value='{{ old(title, \$post->title) }}'

**Security:**

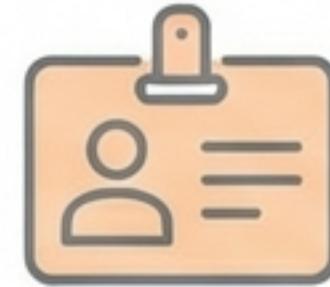
- authorize('update', \$post)

# P-08 セキュリティ & インフラ構成 (Security & Infrastructure)



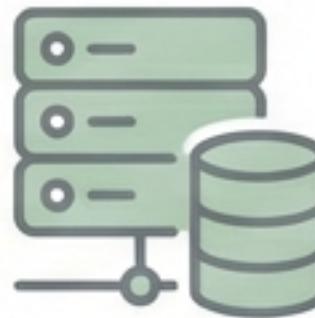
## Authentication

Laravel Breeze  
(Session Mgmt)



## Authorization

Policies  
(Owner-only editing)



## Infrastructure

Docker Compose  
(App/DB)



## CI/Quality

GitHub Actions  
(PHPUnit)

# P-09 結論・まとめ (Summary)

本設計書は、ユーザー中心の記事管理機能と堅牢なセキュリティ要件を両立しています。

-  Authentication: Secure Login/Register/Logout
-  CRUD: Complete Create, Read, Update, Delete flow
-  My Page: Dashboard for self-content management
-  Access Control: Strict Owner-Only editing policies

Ready for implementation with Laravel 12 & Docker.