

# 詳細設計書



# 目次

## 1. プログラム概要

### 1-1 概要

#### 1-1-1 コンポーネント構成

#### 1-1-2 機能構成

#### 1-1-3 論理ER図

### 1-2 規約

#### 1-2-1 コーディング規約

## 2. プログラム詳細

### 2-1 概要

#### 2-1-1 画面一覧

### 2-2 各機能の内部設計詳細

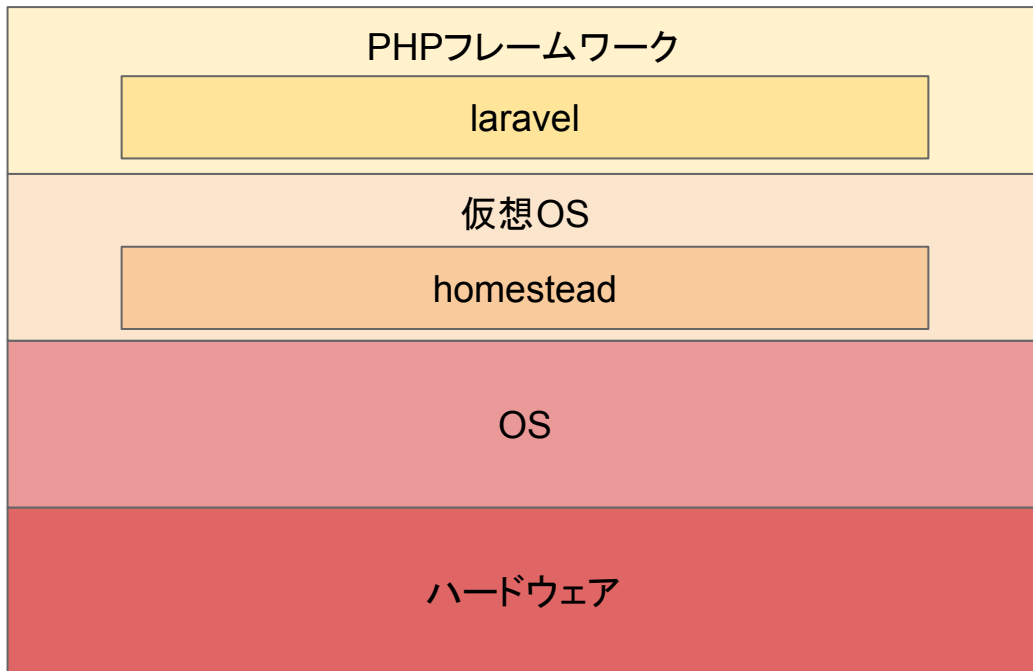
#### 2-2-1 テーブル一覧

#### 2-2-2 ディレクトリ構成

#### 2-2-3 命名規約・ルール

#### 2-2-4 バリデーション設定

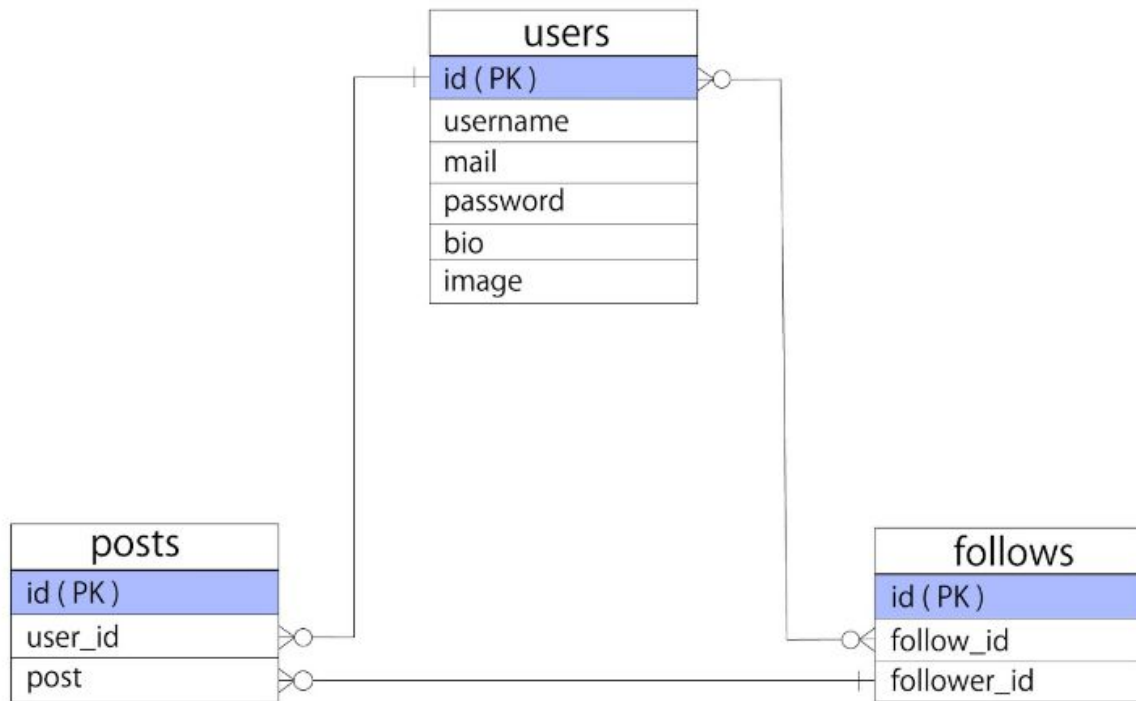
# 1. プログラム概要 | コンポーネント構成



# 1.プログラム概要 | 機能構成

#	機能名	機能名 2	機能概要
1	ログイン機能		入力項目に入れたがテーブルのレコードと一致したらログインして、トップページへ移動する
2	ユーザー登録機能	新規登録機能	新しくusersテーブルに、ユーザー情報を追加する 成功したら、登録完了ページへ移動する
		登録完了機能	ユーザー登録ページから、登録されたユーザーの名前を持ってきて表示させる
3	投稿機能	投稿機能	投稿したい内容をusersテーブルに追加させる
		投稿一覧機能	ログインユーザーとフォローユーザーの投稿一覧を表示させる
		投稿削除機能	選んだログインユーザーの投稿内容を削除する
		投稿編集機能	選んだログインユーザーの投稿内容を編集する
4	プロフィール機能	プロフィール編集機能	ログインユーザーの情報を表示し、フォームを使って情報更新を行えるようにする
		ユーザー情報表示機能	選択したフォロー、フォロワーのユーザー名と自己紹介を表示する
		ユーザーの投稿一覧表示機能	選択したフォロー、フォロワーユーザーの投稿を日時が新しい順で表示させる
5	ユーザー検索機能	検索機能	入力欄に入れたワードをもとに、あいまい検索でユーザー検索ができるようにする
		ユーザー一覧表示機能	SNSを利用している全ユーザーを表示させる
		検索ワード表示機能	検索に使ったワードを、次の検索結果画面で表示させる
		ユーザー検索一覧表示機能	あいまい検索でヒットしたユーザーの一覧を表示させる
6	フォロー機能	フォロー機能	ユーザー一覧表示機能/ユーザー検索一覧表示機能にある未フォローのユーザーをフォローする。
		リムーブ機能	フォローユーザー一覧からの削除をする
		フォローユーザー一覧表示機能	ログインしているユーザーがフォローしているユーザーのみ、アイコンを表示していく
		フォローユーザーの投稿一覧機能	ログインしているユーザーがフォローしているユーザーのみ、投稿日時が新しい順で投稿内容を表示させる
7	フォロワー機能	フォロワーユーザーのアイコン一覧表示機能	ログインしているユーザーがフォローされているユーザーのみ、アイコンを表示していく
		フォロワーユーザーの投稿一覧機能	ログインしているユーザーがフォローされているユーザーのみ、投稿日時が新しい順で投稿内容を表示させる

# 1. プログラム概要 | 論理ER図



# 1. プログラム概要 | コーディング規約

## 目的

└この文書はSNS作成において留意すべき事項にまとめたものである。

## 命名規約

└【プログラム詳細 | 命名規約・ルール】を参考にする事

## コーディングスタイル

└原則、MSDNのサンプルコードを参考にする事、また仕様エディタのフォーマッタ機能によるスタイルを変更しないこと

# 1.プログラム概要 | コーディング規約

## 禁止事項

└ ネットで検索したサンプルプログラムのコードをむやみにコピーして使わないこと(理由:信頼性の欠如)

サンプルプログラムをコピーして使用する場合は、出所をコメントとして明示する。またサンプルプログラムの動作確認を単体で行うこと。

## 2.プログラム詳細 | 画面一覧

画面一覧をAdobeXDにて確認

<https://xd.adobe.com/view/7a1220fc-91b5-4f9a-686a-9ea4fc449dc8-0c23/grid>

PASS:DawnSNS00



## 2.プログラム詳細 | テーブル一覧

テーブル定義							
DB名	dawnsns		文字コード	utf8-general_ci			
テーブル名	users						
	COLUMN	役割	TYPE	NULL	DEFAULT	KEY	OPTION
	id	ID番号	int(11)	not		primary	auto_increment
	username	ユーザー名	varchar(255)	not			
	mail	メールアドレス	varchar(255)	not		unique	
	password	ログインパス	varchar(255)	not		unique	
	bio	自己紹介文	varchar(400)	yes			
	image	ユーザーアイコン	varchar(255)	yes	dawn.png		
	created_at	登録日	timestamp	not	current_timestamp		
	modified_at	更新日	timestamp	not	current_timestamp		on update current_timestamp
	posts						
	COLUMN	役割	TYPE	NULL	DEFAULT	KEY	OPTION
	id	ID番号	int(11)	not		primary	auto_increment
	user_id	だれの投稿か	int(11)				
	post	投稿内容	varchar(400)				
	created_at	登録日	timestamp	not	current_timestamp		
	modified_at	更新日	timestamp	not	current_timestamp		on update current_timestamp
	follows						
	COLUMN	役割	TYPE	NULL	DEFAULT	KEY	OPTION
	id	ID番号	int(11)	not		primary	auto_increment
	follow_id	フォローされた人	int(11)	not			
	follower_id	フォローをした人	int(11)	not			
	created_at	登録日	timestamp	not	current_timestamp		

## 2.プログラム詳細 | ディレクトリ構成

Laravel ver5.5

フォルダ名	役割	ディレクトリ
Controllers	コントローラー(MVC)ファイルの格納場所	app/http/Controllers/
-	モデル(MVC)ファイルの格納場所	app/
-	ビュー(MVC)ファイルの格納場所	resources/views/コントローラーの名前/
icons	ユーザーアイコン画像の格納場所	public/images/icons/
css	CSSファイルの格納場所 適用は3つ reset.css / style.css / logout.css	public/css/
js	JavaScript, JQueryファイルの格納場所	public/js/

## 2. プログラム詳細 | 命名規約・ルール

フロントエンド	命名規約・ルール	例
クラス属性	<ul style="list-style-type: none"><li>・ ケバブケース</li><li>・ クラス属性は、主にCSSの適用に使う</li></ul>	<pre>&lt;p class="dawn"&gt; &lt;p class="dawn-sns-message"&gt;</pre>
id属性	<ul style="list-style-type: none"><li>・ ケバブケース</li><li>・ CSS以外にも、jQueryを使ったアニメーション機能として利用する場合は、複数個所に必要となる以外ではなるべくid属性をつけるようにする</li></ul>	<pre>&lt;p id="Dawn"&gt; &lt;p id="DawnSnsModal"&gt;</pre>
bladeファイル	<ul style="list-style-type: none"><li>・ コントローラーの名前のフォルダの中に配置</li><li>・ ファイル名はコントローラー内のメソッド名と同じとする</li></ul>	UsersController.phpのindexメソッド → resouces/views/users/index.blade.php
bladeの親テンプレート	<ul style="list-style-type: none"><li>・ app.blade.phpでresouces/view/の直下に配置</li><li>・ ヘッダーをここに置く</li></ul>	resouces/views/app.blade.php
フォーム内のname属性	<ul style="list-style-type: none"><li>・ 基本はケバブケース</li><li>・ テーブルと関わるものは、カラムと同じ名前にする</li></ul>	<pre>&lt;input name="dawn-sns-message"&gt; postsテーブルのuser_idに登録させる場合 → &lt;input name="user_id"&gt;</pre>
サーバーサイド	命名規約	例
テーブル名	<ul style="list-style-type: none"><li>・ 小文字/複数形</li></ul>	users / posts / follows
カラム名	<ul style="list-style-type: none"><li>・ 小文字/単数形</li><li>・ JOINで利用する場合に限り、スネークケース</li></ul>	user_id / follow_id / follower_id
コントローラー	<ul style="list-style-type: none"><li>・ キャメルケースの複数形</li></ul>	UserController.php
メソッド	<ul style="list-style-type: none"><li>・ ローワーキャメルケース(最初だけ小文字)の単数形</li></ul>	public function index( ) public function userProfile( )
モデル	<ul style="list-style-type: none"><li>・ キャメルケースの単体系</li></ul>	User.php

## 2.プログラム詳細 | バリデーション設定

本プロジェクトにおけるバリデーションとは、フォームに入力された内容が要件を満たしているかの入力チェックのことを指す

### ユーザー登録機能

入力値フォーム	項目
UserName	<ul style="list-style-type: none"><li>入力必須</li><li>4文字以上,12文字以内</li></ul>
MailAdress	<ul style="list-style-type: none"><li>入力必須</li><li>4文字以上,12文字以内</li><li>登録済みアドレス使用不可</li></ul>
Password	<ul style="list-style-type: none"><li>入力必須</li><li>英数字のみ</li><li>4文字以上,12文字以内</li><li>登録済みPass使用不可</li></ul>
Password confirm	<ul style="list-style-type: none"><li>入力必須</li><li>英数字のみ</li><li>4文字以上,12文字以内</li><li>登録済みPass使用不可</li><li>Password入力欄と一致必須</li></ul>

### プロフィール編集

入力フォーム	項目
UserName	<ul style="list-style-type: none"><li>4文字以上,12文字以内</li></ul>
MailAdress	<ul style="list-style-type: none"><li>4文字以上,12文字以内</li><li>登録済みアドレス使用不可</li></ul> ※アドレスに変更がなければ再登録可能とする
new Password	<ul style="list-style-type: none"><li>入力必須</li><li>英数字のみ</li><li>4文字以上,12文字以内</li><li>登録済みPass使用不可</li><li>Password入力欄と不一致が必須</li></ul>
Bio	<ul style="list-style-type: none"><li>200文字以内</li></ul>
Icon Image	<ul style="list-style-type: none"><li>ファイル名が英数字のみ</li><li>画像(jpg、png、bmp、gif、svg)ファイル以外は不可</li></ul>