

高知工科大学様
大学掲示板 (KUTBBS)
内部設計書 v1.0

グループ 10

Pathfinder

平成 30 年 12 月 10 日

目次

1	システム概要	4
2	動作環境	4
3	開発環境	5
4	内部設計書作成方針	5
4.1	コーディング規約	5
4.1.1	命名規約	5
4.1.2	コーディングスタイル	6
4.2	設計書作成環境	6
4.3	サーバ環境	6
5	データベース設計	7
5.1	データテーブル構成	7
5.2	テーブル設計	8
5.2.1	ユーザテーブル (user)	8
5.2.2	スレッドテーブル (thread)	10
5.2.3	レステーブル (res)	11
5.2.4	スレッドカテゴリテーブル (thread_category)	12
5.2.5	お気に入り掲示板テーブル (favorite_bbs)	12
5.2.6	コレクトユーザテーブル (collect_user)	13
5.2.7	不適切な単語登録テーブル (inadequacy_word)	13
5.2.8	通報テーブル (report)	14
5.2.9	凍結ユーザテーブル (suspend)	15
5.2.10	お知らせテーブル (news)	15
5.2.11	警告注意テーブル (warned_caution)	16
6	ルーティング及び MVC 一覧	16
6.1	ルーティング一覧	17
6.2	ルーティング (管理者側) 一覧	17
6.3	Controller	18
6.3.1	sessions_controller.rb	18
6.3.2	passwords_controller.rb	19
6.3.3	home_controller.rb	19
6.3.4	users_controller.rb	19
6.3.5	results_controller.rb	19
6.3.6	results_categories_controller.rb	20
6.3.7	informations_controller.rb	20
6.3.8	categories_controller.rb	20
6.3.9	threads_controller.rb	20
6.3.10	contents_controller.rb	20
6.4	Controller 層 (管理者側)(仮)	21

6.4.1	admins_session_controller.rb	21
6.4.2	admins_top_controller.rb	22
6.4.3	ng_word_controller.rb	22
6.4.4	manage_controller.rb	22
6.4.5	manage_user_controller.rb	22
6.4.6	announce_controller.rb	23
6.4.7	report_controller.rb	23
6.5	View 層	23
6.5.1	sessions/new.html.erb	23
6.5.2	passwords/new.html.erb	23
6.5.3	home_pages/home.html.erb	24
6.5.4	users/show.html.erb	24
6.5.5	informations/show.html.erb	24
6.5.6	categories/index.html.erb	24
6.5.7	threads/new.html.erb	25
6.5.8	threads/check.html.erb	25
6.5.9	contents/bbs.html.erb	25
6.5.10	contents/check.html.erb	26
6.5.11	contents/report.html.erb	26
6.5.12	contents/check.html.erb	26
6.6	Model 層	26
6.6.1	tb_user.rb	26
6.6.2	tb_thread.rb	27
6.6.3	tb_res.rb	27
6.6.4	tb_thread_category.rb	27
6.6.5	tb_favorite_bbs.rb	27
6.6.6	tb_collect_user.rb	27
6.6.7	tb_inadequacy_word.rb	28
6.6.8	tb_report.rb	28
6.6.9	tb_suspend.rb	28
6.6.10	tb_news.rb	28
6.6.11	tb_warned_caution.rb	28

1 システム概要

本システムは本校の学生同士によって問題提起から問題解決までを匿名で行えるようにするためのシステムである。

本システムは Web ブラウザ上で「掲示板システム」「ユーザシステム」「管理者システム」によって構築される各要素の主な機能は以下に示す。

- 掲示板システム
 - － 掲示板サブシステム
 - － 検索サブシステム
 - － コレクトボタンサブシステム
 - － 通報サブシステム
 - － 通知サブシステム
- ユーザシステム
 - － アカウント登録サブシステム
 - － ログインサブシステム
 - － マイページサブシステム
 - － ブックマークサブシステム
 - － お知らせ表示サブシステム
- 管理者システム
 - － 管理者ログインサブシステム
 - － 子管理者管理サブシステム
 - － お知らせ編集サブシステム
 - － 掲示板編集サブシステム
 - － ユーザ管理サブシステム

2 動作環境

本システムの動作環境は以下の通りである。

- 動作環境
 - － CPU : ARM Cortex-A53 1GHz 以上
 - － GPU : Broadcom VideoCore IV
 - － メモリ : 2GB 以上
 - － ストレージ : 4GB eMMC / SD カード PIN
 - － OS
 - * Linux version

- * ubuntu version
- * MacOS version
- * Windows 7 , 8 , 10
- * iOS 10.0
- web サーバ : AWS EC2
- App サーバ : Ruby on Rails version 5.1.4
- RDBMS : MySQL version 8.0
- 使用ブラウザ : GoogleChrome62.0 , Firefox version 57.0

3 開発環境

本システムの開発環境は以下の通りである。

- OS : Linux , ubuntu , MacOS , Windows , iOS10.0
- HTML : version 5
- 使用言語
 - ruby version 2.4.2
 - Ruby on Rails version 5.1.4
 - CSS
 - JavaScript
- サーバ :
- データベース : MySQL version 8.0
- web サーバ : AmazonWebServices

4 内部設計書作成方針

4.1 コーディング規約

4.1.1 命名規約

- 変数名・メソッド名
 - 小文字始まりとする
 - 基本的に意味のある単語を使用する
 -
- 定数
 - 全て大文字を使用する

- クラス名・構造体名
 - 変数名・メソッド名を同様に意味のある単語を使用する
 - 大文字始まりとする
 - 複数の単語を組み合わせる際は先頭文字を大文字で表記する

4.1.2 コーディングスタイル

- インデント
 - インデントにはタブを使用する（半角スペース 4 文字）
- 括弧
 - 中括弧は改行して始める
 - 小括弧の前後にはスペースを使用しない
- 演算子
 - 演算子の前後には半角スペースを一文字使用する

4.2 設計書作成環境

内部設計書の作成環境は、表 1 に示します。

表 1: 内部設計書の作成環境環境

組版処理システム	LATEX,dvpdfmx
文字コード	UTF-8
改行コード	LF(0x0A)

4.3 サーバ環境

本システムを利用するためには Amazon Web Services(AWS) の EC2 インスタンスを用いて実現します。サーバ環境は表 2 に示します。

表 2: サーバ側の動作環境

対応 OS	Ruby on Rails
vCPU	1
メモリ (GiB)	1
ストレージ	30GB

5 データベース設計

本章では本システムにおけるデータベース MySQL についてを示す。

5.1 データテーブル構成

本小節ではデータテーブルの構成を ER 図を用いて示す。

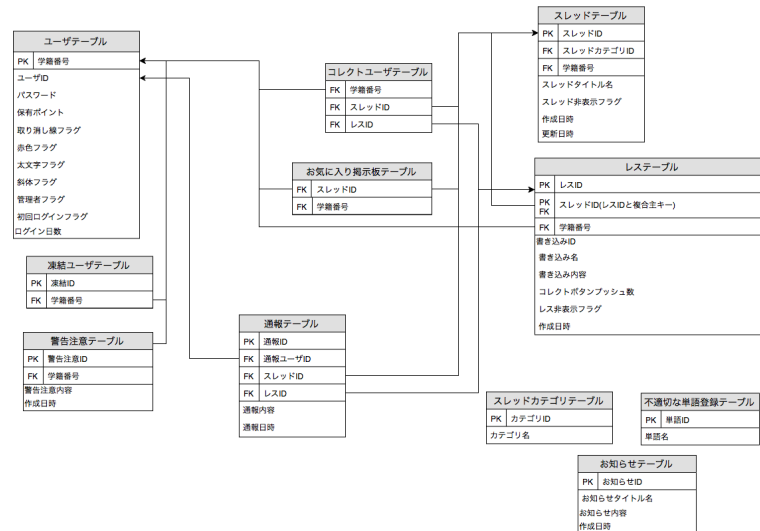


図 1: ER 図

5.2 テーブル設計

本小節ではデータベース構成するテーブルについて示す。また、各カラムについても詳細も示す。

5.2.1 ユーザテーブル (user)

ユーザテーブルではユーザに関する情報を管理する。このテーブルの詳細は表 3 に示す。

表 3: ユーザテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
学籍番号	student_id	char	7	×	-	PK
ユーザ ID	user_id	varchar	20	×	-	-
パスワード	password	varchar	20	×	-	-
保有ポイント	now_point	int	4	×	-	-
取り消しフラグ	cancel_flag	int	1	×	-	-
赤色フラグ	color_flag	int	1	×	-	-
斜体フラグ	diagonal_flag	int	1	×	-	-
太文字フラグ	bold_letters_flag	int	1	×	-	-
管理者フラグ	administrator_flag	int	1	○	-	-
初回ログインフラグ	first_login_flag	int	1	×	-	-
ログイン日数	count_login	int	4	×	-	-

- 学籍番号
ユーザの学籍番号を示す値であり、自テーブルの主キーである。NULL 値は含まない。値は固定 7 文字の半角英数字にて構成される。
- ユーザ ID
ユーザを識別するための ID を示す値である。NULL 値は含まない。値は 20 文字以下の半角英数字にて構成される。
- パスワード
ユーザを識別するためのパスワードを示す値である。NULL 値は含まない。値は 20 文字以下の半角英数字にて構成される。
- 保有ポイント
ユーザがログインした日数とコレクトボタンを 10 回以上押されたことで獲得したポイントを示す値である。NULL 値は含まない。4 桁 (0001 から 9999) の数値にて構成される。
- 取り消しフラグ
拡張機能の 1 つであり、取り消し線を付与することを示す値である。NULL 値は含まない。値は 1 桁の 0(OFF)、1(ON) の数値にて構成される。
- 赤色フラグ
拡張機能の 1 つであり、赤色を付与することを示す値である。NULL 値は含まない。値は 1 桁の 0(OFF)、1(ON) の数値にて構成される。

- 斜体フラグ
拡張機能の1つであり、斜体を付与することを示す値である。NULL 値は含まない。値は1桁の0(OFF)、1(ON)の数値にて構成される。
- 太文字フラグ
拡張機能の1つであり、太文字を付与することを示す値である。NULL 値は含まない。値は1桁の0(OFF)、1(ON)の数値にて構成される。
- 管理者フラグ
親管理者と子管理者を識別することを示す値である。値は1桁の0(子管理者)、1(親管理者)の数値にて構成される。
- 初回ログインフラグ
初回ログインを識別することを示す値である。NULL 値は含まない。値は1桁の0(OFF)、1(ON)の数値にて構成される。
- ログイン日数
ユーザがログインをした日数を記録することを示す値である。NULL 値は含まない。値は4桁(0001 から 9999)の数値にて構築される。

5.2.2 スレッドテーブル (thread)

スレッドテーブルではスレッドに関する情報を管理する。このテーブルの詳細は表 4 に示す。

表 4: スレッドテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	PK
スレッドカテゴリ ID	thread_category_id	int	3	×	unsigned zerofill auto_increment	FK:thread_category
学籍番号	student_id	char	7	×	-	FK:user
スレッドタイトル名	thread_category_name	varchar	50	×	-	-
スレッド非表示フラグ	thread_hide_flag	int	1	×	-	-
作成日時	created_at	timestamp	-	×	default current_timestamp	-
更新日時	updated_at	timestamp	-	×	default current_timestamp on update current_timestamp	-

- スレッド ID
自テーブルの主キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。
- スレッドカテゴリ ID
スレッドカテゴリテーブルを参照する際の外部キーである。NULL 値は含まない。値は 3 桁 (001 から 999) の数値にて構成され自動追加される。
- 学籍番号
ユーザテーブルを参照する際の外部キーである。NULL 値は含まない。固定 7 文字の半角英数字にて構成される。
- スレッドタイトル名
スレッドタイトルを示す値である。NULL 値は含まない。値は 50 文字以下の文字列にて構成される。
- スレッド非表示フラグ
管理者の操作権限でスレッドを非表示を示す値である。NULL 値は含まない。値は 1 桁の数値で 0(OFF)、1(ON) の数値にて構成される。
例としてスレッド非表示フラグを 1 にした場合、ユーザ側からはスレッドが非表示になる。
- 作成日時
レコードを作成した日付・時刻を示す値である。NULL 値は含まない。レコードが作成されるたびに自動追加を行う。
- 更新日時
レコードを更新した日付・時刻を示す値である。NULL 値は含まない。レコードが更新されるたびに自動更新を行う。

5.2.3 レステータブル (res)

レステータブルではレスに関する情報を管理する。このテーブルの詳細は表 5 に示す。

表 5: レステータブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
レス ID	res_id	int	4	×	unsigned zerofill auto_increment	PK
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	PK(res 複合) FK:thread
学籍番号	student_id	char	7	×	-	FK:user
書き込み ID	write_id	char	8	×	-	-
書き込み名	write_name	varchar	15	×	-	-
書き込み内容	write_content	varchar	200	×	-	-
コレクトプッシュ数	collect_push_count	int	4	×	-	-
書き込み ID	write_id	char	8	×	-	-
レス非表示フラグ	res_hide_flag	int	1	×	-	-
作成日時	created_at	timestamp	-	×	default current_timestamp	-

- レス ID
自テーブルの主キーである。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成され自動追加される。
- スレッド ID
スレッドテーブルを参照する際の外部キーであり、スレ ID との複合主キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。
- 学籍番号
ユーザーテーブルを参照する際の外部キーである。NULL 値は含まない。固定 7 文字の半角英数字にて構成される。
- 書き込み ID
ユーザが書き込みを行う際に表示する ID を示す値である。NULL 値は含まない。値は固定 8 文字の半角英数字にて構成される。
- 書き込み名
書き込みを行う際に表示する名前を示す値である。NULL 値は含まない。値は 15 文字以下の文字列にて構成される。
- 書き込み内容
書き込み内容を示す値である。NULL 値は含まない。値は 200 文字以下の文字列にて構成される。
- コレクトボタンプッシュ数
レスに対してコレクトボタンが押された回数を示す値である。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成される。

- レス非表示フラグ

管理者の操作権限でレスを非表示を示す値である。NULL 値は含まない。値は 1 桁の数値で 0(OFF)、1(ON) の数値にて構成される。

例としてレス非表示フラグを 1 にした場合、ユーザ側からはレスが非表示になる。

- 作成日時

レコードを作成した日付・時刻を示す値である。NULL 値は含まない。レコードが作成されるたびに自動追加される。

5.2.4 スレッドカテゴリテーブル (thread_category)

スレッドカテゴリテーブルは各カテゴリごとに分けられたスレッドの情報を管理する。このテーブルの詳細は表 6 に示す。

表 6: スレッドカテゴリテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
カテゴリ ID	category_no	int	3	×	unsigned zerofill auto_increment	PK
カテゴリ名	category_name	varchar	15	×	-	-

- カテゴリ ID

自テーブルの主キーである。NULL 値は含まない。値は 3 桁 (001 から 999) の数値にて構築され自動追加される。

- カテゴリ名

カテゴリの名前を示す値である。NULL 値は含まない。値は 15 文字以下の文字列にて構成される。

5.2.5 お気に入り掲示板テーブル (favorite_bbs)

お気に入り掲示板テーブルではブックマークとして登録したスレッド情報を管理する。このテーブルの詳細は表 7 に示す。

表 7: お気に入り掲示板テーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
学籍番号	student_id	char	7	×	-	FK:user
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	FK:thread

- 学籍番号

ユーザテーブルを参照する際の外部キーである。NULL 値は含まない。固定 7 文字の半角英数字にて構成される。

- スレッド ID

スレッドテーブルを参照する際の外部キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構築され自動追加される。

5.2.6 コレクトユーザテーブル (collect_user)

コレクトユーザテーブルではコレクトボタンを押されたことに関する情報を管理する。このテーブルの詳細は表 8 に示す。

表 8: コレクトユーザテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
学籍番号	student_id	char	7	×	-	FK:user
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	FK:thread
レス ID	res_no	int	4	×	unsigned zerofill auto_increment	FK:res

- 学籍番号

ユーザテーブルを参照する際の外部キーである。NULL 値は含まない。固定 7 文字の半角英数字にて構成される。

- スレッド ID

スレッドテーブルを参照する際の外部キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。

- レス ID

レステーブルを参照する際の外部キーである。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成され自動追加される。

5.2.7 不適切な単語登録テーブル (inadequacy_word)

不適切な単語登録テーブルでは管理者が誹謗中傷や公序良俗に違反していると考えられる単語を登録した情報を管理する。このテーブルの詳細は表 9 に示す。

表 9: 不適切な単語登録テーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
単語 ID	word_id	int	5	×	unsigned zerofill auto_increment	PK
単語名	word_name	varchar	200	×	-	-

- 単語 ID

自テーブルの主キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。

- 単語名

管理者の操作権限で登録した単語を示す値である。NULL 値は含まない。値は 200 文字以下の文字列にて構成される。

5.2.8 通報テーブル (report)

通報テーブルではユーザが誹謗中傷や公序良俗に違反するなどの不適切な内容であると判断したスレッドまたはレスを管理者に通報した時の情報を管理する。また、通報した内容の情報も管理する。このテーブルの詳細は表 10 に示す。

表 10: 通報テーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
通報 ID	report_id	int	5	×	unsigned zerofill auto_increment	PK
通報ユーザ ID	report_user_id	varchar	20	×	-	FK:user
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	FK:thread
レス ID	res_id	int	4	×	unsigned zerofill auto_increment	FK:res
通報内容	report_content	varchar	200	×	-	-
通報日時	report_at	timestamp	-	×	default current_timestamp	-

- 通報 ID

自テーブルの主キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。

- 通報ユーザ ID

ユーザテーブルを参照する際の外部キーである。NULL 値は含まない。値は 20 文字以下の半角英数字にて構成される。

- スレッド ID

スレッドテーブルを参照する際の外部キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。

- レス ID

レステーブルを参照する際の外部キーである。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成され自動追加される。

- 通報内容

通報内容を示す値である。NULL 値は含まない。値は 200 文字以下の文字列にて構成される。

- 通報日時

レコードを作成した日付・時刻を示す値である。NULL 値は含まない。レコードが作成されるたびに自動追加を行う。

5.2.9 凍結ユーザテーブル (suspend)

凍結ユーザテーブルでは迷惑行為が改善されないユーザのアカウントを凍結した情報を管理する。このテーブルの詳細は表 11 に示す。

表 11: 凍結ユーザテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
凍結 ID	suspend_id	int	4	×	unsigned zerofill auto_increment	PK
学籍番号	student_id	char	7	×	-	FK:user

- 凍結 ID
自テーブルの主キーである。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成され自動追加される。
- 学籍番号
ユーザテーブルを参照する際の外部キーである。。NULL 値は含まない。値は固定 7 文字の半角英数字にて構成される。

5.2.10 お知らせテーブル (news)

お知らせテーブルではお知らせに関する情報を管理する。このテーブルの詳細は表 12 に示す。

表 12: お知らせテーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
お知らせ ID	news_id	int	5	×	unsigned zerofill auto_increment	PK
お知らせタイトル名	news_title	varchar	50	×	-	-
お知らせ内容	news_title	varchar	400	×	-	-
作成日時	created_at	timestamp	-	×	default current_timestamp	-

- お知らせ ID
自テーブルの主キーである。NULL 値は含まない。値は 5 桁 (00001 から 99999) の数値にて構成され自動追加される。
- お知らせタイトル名
お知らせタイトルを示す値である。NULL 値は含まない。値は 50 文字以下の文字列にて構成される。
- お知らせ内容
お知らせ内容を示す値である。NULL 値は含まない。値は 400 文字以下の文字列にて構成される。
- 作成日時
レコードを作成した日付・時刻を示す値である。NULL 値は含まない。レコードが作成されるたびに自動追加を行う。

5.2.11 警告注意テーブル (warned_caution)

警告テーブルでは管理者がユーザに対しての警告・注意喚起に関する情報を管理する。このテーブルの詳細は表 13 に示す。

表 13: 警告注意テーブル

論理名	物理名	データ型	精度	NULL	オプション	PK/FK:mode
警告注意 ID	warned_caution_id	int	4	×	unsigned zerofill auto_increment	PK
学籍番号	student_id	char	7	×	-	FK:user
警告注意タイトル名	warned_caution_title	varchar	50	×	-	-
警告注意内容	warned_caution_title	varchar	400	×	-	-
作成日時	created_at	timestamp	-	×	default current_timestamp	-

- 警告注意 ID
自テーブルの主キーである。NULL 値は含まない。値は 4 桁 (0001 から 9999) の数値にて構成され自動追加される。
- 学籍番号
ユーザテーブルを参照する際の外部キーである。NULL 値は含まない。値は固定 7 文字の半角英数字にて構成される。
- 警告注意タイトル名
警告注意タイトルを示す値である。NULL 値は含まない。値は 50 文字以下の文字列にて構成される。
- 警告注意内容
警告注意内容を示す値である。NULL 値は含まない。値は 400 文字以下の文字列にて構成される。
- 作成日時
レコードを作成した日付・時刻を示す値である。NULL 値は含まない。レコードが作成されるたびに自動追加を行う。

6 ルーティング及び MVC 一覧

この章では、Rails の規約に従った URL 規則を示す。また、HTTP メソッドと URL によって呼び出される Controller と Action を示す。さらに、View 及び Model も示す。

6.1 ルーティング一覧

以下の表は、Rails の規則に従った URL 規則の表である。また、Controller とその Action についても示す。

表 14: ルーティング一覧

No.	URL	METHOD	Controller#Action
1	/Login	GET	sessions#new
2		POST	sessions#create
3	/Logout	DELETE	sessions#destroy
4	/passwords/:id	GET	passwords#new
5	/passwords/:id	PATCH	passwords#change
6	/home	GET	home_pages/#home
7	/home	PATCH	home_pages/#threads_hide
8	/users/:id	GET	users#show
9	/users/:id	PATCH	users#update
10	/results/:title	GET	results#search
11	/results/:title/categories/:id	GET	categories#search
12	/informations/:id	GET	informations#show
13	/categories/:id	GET	categories#index
14	/threads/new	GET	threads#new
15	/threads/check	GET	threads#check
16	/threads	POST	threads#create
17	/threads/:id/report_new	GET	threads#report_new
18	/threads/check	GET	threads#check
19	/threads	POST	threads#create
20	/threads/:id/contents	GET	contents#bbs
21	/threads/:id/contents/check	GET	contents#check
22	/threads/:id/contents	POST	contents#write
23	/threads/:id/contents	PATCH	contents#responses_hide
24	/threads/:id/contents/:id/report	GET	contents#report
25	/threads/:id/contents/:id/check	GET	contents#check
26	/threads/:id/contents	POST	contents#report_create

6.2 ルーティング (管理者側) 一覧

(v2 でユーザ側と管理者側を統合・修正します。)

呼び出された URL と HTTP メソッドによって Rails で呼び出す Controller のアクションを定義する。

表 15: ルーティング (管理者側) 一覧

No.	URL	METHOD	Controller#Action
1	admins/login	GET	admins_session
2	admins/login	POST	admins_session#login
3	admins/logout	POST	admins_session#logout
4	admins/top	GET	admins_top#home
5	admins/ng_word	GET	ng_word#home
6	admins/ng_word/create	POST	ng_word#create
7	admins/ng_word/destroy	DELETE	ng_word#destroy
8	admins/manage	GET	manage#home
9	admins/manage/signup	GET	manage#signup
10	admins/manage/create	POST	manage#create
11	admins/manage/ok	GET	manage#ok
12	admins/manage/destroy_check	GET	manage#destroy_check
13	admins/manage/destroy	DELETE	manage#destroy
14	admins/manage_user	GET	manage_user#home
15	admins/manege_user/signup	GET	manage_user#user_signup
16	admins/manage_user/create	POST	manage_user#user_create
17	admins/manage_user/search	GET	manage_user#search
18	admins/manage_user/:id	GET	manage_user#info
19	admins/manage_user/warning_edit	GET	manage_user#warning_edit
20	admins/manage_user/warning_check	GET	manage_user#warning_check
21	admins/manage_user/warning_send	POST	manage_user#warning_send
22	admins/manage_user/ban	PATCH	manage_user#ban
23	admins/announce	GET	announce#edit
24	admins/announce_check	GET	announce#check
25	admins/announce_send	POST	announce#send
26	admins/report	GET	report#home
27	admins/report/hide	PATCH	report#hide

6.3 Controller

6.3.1 sessions_controller.rb

名称：セッション情報処理

処理：

-new: ログインフォームである sessions_new.html.erb を表示させる。

-create: ユーザの ID とパスワードで認証を行い、認証されるとそのユーザのセッションを作成する。セッションを作成したユーザによって、以下のルーティングにリクエストを行う。

ユーザが管理者:/admin に対して GET メソッドでルーティングにリクエストを行う。

ユーザが一般ユーザ:/home に対して GET メソッドでルーティングにリクエストを行う。また、そのユーザが初回ログインの場合は、/passwords に GET メソッドでルーティングにリクエストを行う。なお、認証が失敗

した場合は再度/Login に対して GET メソッドでルーティングを行う。

-destroy:作成したユーザのセッションを破棄する。

6.3.2 passwords_controller.rb

名称:パスワード変更処理

処理:

-new:初回ログインを行なった該当するユーザに対して、パスワード変更用フォームである passwords_new.html.erb を表示させる。

-change:あらかじめ登録されていたユーザのパスワード情報を、変更用フォームから入力された新しいパスワードに更新する。更新後、/home に対して GET メソッドでルーティングにリクエストする。

6.3.3 home_controller.rb

名称:ホーム画面処理

処理:

-home:ホーム画面である home.html.erb を表示する。スレッドタイトル入力テキストボックスに検索したいスレッドタイトルを入力し、検索ボタンを押すと、/results/:id に GET メソッドでルーティングにリクエストを行う。お知らせを押すと、/infomations/:id に GET メソッドでルーティングを行う。マイページボタンを押すと、/user/:id に GET メソッドでルーティングを行う。

-threads_hide:不適切なスレッドを非表示化する。この処理は、管理者以外行うことはできない (非表示化ボタン自体が存在しない)。このアクションを行うと、/home に対して PATCH メソッドでルーティングにリクエストする。

6.3.4 users_controller.rb

名称:マイページ画面処理

処理:

-show:マイページ画面である show.html.erb を表示する。

-update:拡張機能を開放する (User テーブルにおけるユーザの各種拡張フラグを 1 に更新する) アクションである。解放後、/user/:id に PATCH メソッドでルーティングにリクエストする。

6.3.5 results_controller.rb

名称:スレッドタイトル検索処理

処理:

-search:スレッドタイトル検索機能の処理を行うアクションである。検索を行い、該当するスレッドを取得した後は、そのスレッドを一覧として表示する (これが search.html.erb となる)。表示されたスレッドタイトルを押すと、/threads/:id/contents に GET メソッドでルーティングにリクエストする。

6.3.6 results_categories_controller.rb

名称：カテゴリ別スレッドタイトル検索処理

処理:

-search:スレッドタイトル検索機能の処理を行うアクションである。先ほど述べたスレッドタイトル検索処理とほとんど同じであるが、スレッドタイトル以外にも、該当するカテゴリであるかどうかを条件として検索を行う処理である。検索条件以外の処理に違いは存在しない。

6.3.7 informations_controller.rb

名称：お知らせ画面処理

処理:

-show:お知らせ詳細画面である show.html.erb を表示する。

6.3.8 categories_controller.rb

名称：カテゴリ別トップ画面処理

処理:

-new:カテゴリ別トップ画面である categories.html.erb を表示する。この画面もトップ画面同様、スレッドタイトル入力テキストボックスが存在しているがこちらから検索を行うと、/results/:title/categories/:id に GET メソッドでルーティングをリクエストする。マイページに遷移するボタンの処理は、トップ画面と同様である。また、スレッドタイトルを押した場合は、/threads/:id/contents に GET メソッドでルーティングをリクエストする。

6.3.9 threads_controller.rb

名称：スレッド作成処理

処理:

-new::スレッドを新規作成するフォームである new.html.erb を表示させる。作成ボタンを押した後は/threads/check に GET メソッドでルーティングをリクエストする。

-check:スレッド作成確認画面である check.html.erb を表示させる。確認ボタンを押すと、/threads に POST メソッドでルーティングをリクエストする。

-create:new アクションで入力されたスレッドタイトルと最初の書き込み内容を反映したスレッドを作成するアクションである。スレッドを作成した後は、カテゴリ別トップページである/categories/:id に GET メソッドでルーティングをリクエストする。

6.3.10 contents_controller.rb

名称：スレッド閲覧処理

処理:

-bbs::スレッドを閲覧するページを表示するアクションである。通報ボタンを押すと、/thread/:id/contents/:id/report

に GET メソッドでルーティングをリクエストする。書き込みボタンを押すと、`/thread/:id/contents/check` に GET メソッドでルーティングをリクエストする。

-check:書き込み確認ページである `contents.check.html.erb` を表示するアクションである。

-write:thread に書き込みを行うアクションである。書き込み確認ページに存在する書き込みボタンを押すところのアクションが実行され、その後`/thread/:id/contents/`に POST メソッドでルーティングをリクエストする。

-responses.hide:このアクションを実行できるのは管理者だけである。このアクションを実行すると、PATCH メソッドで`/thread/:id/contents`にルーティングをリクエストする。

-report:このアクションは、通報用フォームである `contents.html.erb` を表示させるアクションである。必要事項を入力し通報ボタンを押すと、通報確認ページである`/thread/:id/contents/:id/check`に GET メソッドでルーティングをリクエストする。

-check:通報確認ページである `check.html.erb` を表示させるアクションである。

-report_create 通報確認ページで確認ボタンを押すと、通報内容が作成される。その後、POST メソッドで`/thread/:id/contents`にルーティングにリクエストを行う。

6.4 Controller 層 (管理者側)(仮)

6.4.1 admins_session_controller.rb

名称: セッション情報処理

概要: 管理者のセッション情報を処理する

処理:

- home: `admins.html.erb` を表示する。- login: 入力された ID とパスワードを取得する。if 文でデータベース上に存在するか判別する。

存在する場合の処理

- session 変数にデータベース上のユーザ id を代入する。
- flash 変数に「ログインしました」という文字列を代入する。
- `/admins/top` へリダイレクトする。

存在しない場合の処理

- エラーメッセージ用の変数にエラーメッセージを代入する。
- 入力された ID とパスワードを初期値として設定する。
- `/admins/login` へリダイレクトする。

-logout: session 変数に nil を代入する。flash 変数に「ログアウトしました」という文字列を代入する。`/admins/login` へリダイレクトする。

6.4.2 admins_top_controller.rb

名称: 管理者 TOP 画面情報処理

概要: 管理者 TOP 画面情報を処理する

処理:

-home: admins/top.html を表示する。

6.4.3 ng_word_controller.rb

名称: 不適切な単語管理処理

概要: 不適切な単語の登録・削除の処理を行う。

処理:

-home: admins/ng_word.html.erb を表示する。

-create: 入力された文字列を受け取り、受け取った文字列をデータベースに保存する。保存に成功した場合、flash 変数に「単語を登録しました」という文字列を代入し、admins/ng_word へリダイレクトする。

-destroy: 特定の文字列を受け取り、削除処理をする。削除後、flash 変数に「単語を削除しました」という文字列を代入して admins/ng_word へリダイレクトする。

6.4.4 manage_controller.rb

名称: 子管理者管理処理

概要: 子管理者管理の処理をする。

処理:

-home: admins/manage.html.erb を表示する。

-signup: admins/manage/signup.html.erb を表示する。

-create: 子管理者アカウントを発行し、admins/manage/ok へリダイレクトする。

-destroy_check: admins/manage/destroy_check.html.erb を表示する。

-destroy: 特定の子管理者アカウント情報を受け取り、削除処理をする。flash 変数に「子管理者アカウントを削除しました」という文字列を代入し、admins/magnage へリダイレクトする。

6.4.5 manage_user_controller.rb

名称: ユーザ情報管理処理

概要: ユーザ情報管理の処理をする。

処理:

-home: admins/manage_user.html.erb を表示する。

-signup: admins/manage_user/signup.html.erb を表示する。

-create: 受け取った整数の範囲のユーザの ID とパスワードを乱数で生成し、データベースに保存する。保存に成功した場合、flash 変数に「ユーザアカウントを作成しました」という文字列を代入し、admins/manage_user へリダイレクトする。保存に失敗した場合、flash 変数にエラーメッセージを代入し、さらに入力されたされた文字列を初期値として admins/manage_user/signup へリダイレクトする。

-search: admins/manage_user/search.html.erb を表示する。

-info: 特定のユーザの id を受け取り、admins/manage_user/:id.html.erb を表示する。

-warning_edit: admins/manage_user/warning_edit.html.erb を表示する。

- warning_check: admins/manage_user/warning_check.html.erb を表示する。
- warning_send: 入力された文字列を受け取り、データベースに保存する。保存に成功した場合、flash 変数に「警告・注意喚起を送信しました」という文字列を代入し、admins/manage_user/:id へリダイレクトする。
- ban: 特定のアカウントの書き込み禁止の処理を行う。処理後は admin/manage_user/:id へリダイレクトする。

6.4.6 announce_controller.rb

名称: お知らせ情報処理

概要: お知らせ情報の処理を行う。

処理:

-edit: admins/announce.html.erb を表示する。

-check: admins/announce_check.html.erb を表示する。

-send: 作成したお知らせをデータベースに保存する。保存後は admins/announce へリダイレクトする。

6.4.7 report_controller.rb

名称: 通報情報処理

概要: 通報情報の処理を行う。

処理:

-home: admins/report.html.erb を表示する。

-hide: 特定の通報されたスレッドまたはレスを非表示化処理をする。処理後は admins/report へリダイレクトする。

6.5 View 層

(現時点では掲示板の画面詳細のみ記述しています。v2 で管理者側の画面詳細の追記とこの章全体の修正をします。)

6.5.1 sessions/new.html.erb

【名称】

ユーザログイン画面

【概要】

ユーザが ID とパスワードを入力して本システムにログインするための画面である。

【処理フロー】

- ENTER ボタンを押すと、入力された ID とパスワードを情報として/Login に POST リクエストでルーティングにリクエストする。

6.5.2 passwords/new.html.erb

【名称】

新規登録画面

【概要】

初回ログインをしたユーザが ID とパスワードを変更し、本システムに本登録する画面である。

【処理フロー】

- 登録ボタンを押すと、入力された ID とパスワードを情報として /passwords/:id に PATCH メソッドでルーティングにリクエストする。

6.5.3 home_pages/home.html.erb

【名称】

掲示板 TOP 画面【概要】

掲示板 TOP 画面を表示する画面である。

【処理フロー】

- 検索ボタンを押すと、/result/:title に GET メソッドでルーティングにリクエストする。
- カテゴリボタンを押すと、/categories/:id に GET メソッドでルーティングにリクエストする。
- お知らせタイトルを押すと、/information/:id に GET メソッドでルーティングにリクエストする。
- ログアウトボタンを押すと、/Logout に DELETE メソッドでルーティングにリクエストする。

6.5.4 users/show.html.erb

【名称】

マイページ画面

【概要】

ユーザのマイページを表示する画面である。

【処理フロー】

- 拡張機能の開放ボタンを押すと、保有ポイントを情報として、/users/:id に PATCH メソッドでルーティングにリクエストする。

6.5.5 informations/show.html.erb

【名称】

お知らせ画面【概要】

管理者からユーザに対するお知らせの詳細を表示する画面である。

6.5.6 categories/index.html.erb

【名称】

カテゴリ TOP 画面

【概要】

各カテゴリのカテゴリ ID に該当するスレッドを表示する画面である。

【処理フロー】

- スレッドタイトルを押すと、/thread/:id/contents に GET メソッドでルーティングにリクエストする。
- 検索ボタンを押すと、/result/:title/categories/:id に GET メソッドでルーティングにリクエストする。
- スレッド新規作成ボタンを押すと、/thread/new に GET メソッドでルーティングにリクエストする。
- 通報ボタンを押すと、/thread/report_new に GET メソッドでルーティングにリクエストする。

6.5.7 threads/new.html.erb

【名称】

スレッド作成画面

【概要】

スレッドの新規作成フォームを表示する画面である。

【処理フロー】

- 作成ボタンを押すと、/threads/check に GET メソッドでルーティングにリクエストする。

6.5.8 threads/check.html.erb

【名称】

スレッド作成確認画面

【概要】

新規作成するスレッドの内容確認を表示する画面である。

【処理フロー】

- はいボタンを押すと、スレッドタイトル名と書き込み内容を情報として、/threads に POST メソッドでルーティングにリクエストする。
- いいえボタンを押すと、/threads/new に GET メソッドでルーティングにリクエストする。

6.5.9 contents/bbs.html.erb

【名称】

スレッド閲覧画面

【概要】

スレッドの書き込み及び書き込みの入力フォームを表示する画面である。

【処理フロー】

- 送信ボタンを押すと、/threads/:id/contents/check に GET メソッドでルーティングにリクエストする。
- 通報ボタンを押すと、/threads/:id/contents/report に GET メソッドでルーティングにリクエストする。
- 非表示ボタンを押すと、レス ID を情報として、/threads/:id/contents/ に PATCH メソッドでルーティングにリクエストする。

6.5.10 contents/check.html.erb

【名称】

レス書き込み内容確認画面

【概要】

スレッドに書き込むレスの内容確認を表示する画面である。

【処理フロー】

- はいボタンを押すと、書き込み内容を情報として、/contents/write に POST メソッドでルーティングにリクエストする。
- いいえボタンを押すと、/threads/:id/contents/ に GET メソッドでルーティングにリクエストする。

6.5.11 contents/report.html.erb

【名称】

通報画面

【概要】

不適切な書き込みを通報する入力フォームを表示する画面である。

【処理フロー】

- 送信ボタンを押すと、/threads/:id/contents/:id/check に GET メソッドでルーティングにリクエストする。

6.5.12 contents/check.html.erb

【名称】

通報内容確認画面

【概要】

通報の内容確認を表示する画面である。

【処理フロー】

- はいボタンを押すと、通報内容を情報として、/threads/:id/contents に POST メソッドでルーティングにリクエストする。
- いいえボタンを押すと、/threads/:id/contents/:id/report に GET メソッドでルーティングにリクエストする。

6.6 Model 層

6.6.1 tb_user.rb

名称: ユーザ情報管理部

概要: ユーザテーブルの管理を行う。

6.6.2 tb_thread.rb

名称:スレッド情報管理部

概要:スレッドテーブルの管理を行う。

関係:

- スレッド情報管理部 N:1 ユーザ情報管理部
- スレッド情報管理部 N:1 スレッドカテゴリ情報管理部

6.6.3 tb_res.rb

名称:レス情報管理部

概要:レステーブルの管理を行う。

関係:

- レス情報管理部 N:1 スレッド情報管理部
- スレッド情報管理部 N:M ユーザ情報管理部

6.6.4 tb_thread_category.rb

名称:スレッドカテゴリ情報管理部

概要:スレッドカテゴリテーブルの管理を行う。

6.6.5 tb_favorite_bbs.rb

名称:お気に入り掲示板情報管理部

概要:お気に入り掲示板テーブルの管理を行う。

関係:

- お気に入り掲示板情報管理部 0 か N:0 か N ユーザ情報管理部
- お気に入り掲示板情報管理部 N:1 スレッド情報管理部

6.6.6 tb_collect_user.rb

名称:コレクトユーザ情報管理部

概要:コレクトユーザテーブルの管理を行う。

関係:

- コレクトユーザ情報管理部 1:0 か 1 ユーザ情報管理部
- コレクトユーザ情報管理部 N:1 スレッド情報管理部
- コレクトユーザ情報管理部 N:0 か 1 レス情報

6.6.7 tb_inadequacy_word.rb

名称:不適切な単語登録情報管理部

概要:不適切な単語登録テーブルの管理を行う。

6.6.8 tb_report.rb

名称:通報情報管理部

概要:通報テーブルの管理を行う。

関係:

- 通報情報管理部 N:M ユーザ情報管理部
- 通報情報管理部 N:M スレッド情報管理部
- 通報情報管理部 N:M レス情報管理部

6.6.9 tb_suspend.rb

名称:凍結ユーザ情報管理部

概要:凍結ユーザテーブルの管理を行う。

関係:

- 凍結ユーザ情報管理部 1:1 ユーザ情報管理部

6.6.10 tb_news.rb

名称:お知らせ情報管理部

概要:お知らせテーブルの管理を行う。

6.6.11 tb_warned_caution.rb

名称:警告注意情報管理部

概要:警告注意テーブルの管理を行う。

関係:

- 警告注意情報管理部 1:N ユーザ情報管理部