

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

2018 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	テーブル設計	7
5.1	ユーザテーブル (user)	7
5.2	スレッドテーブル (thread)	8
5.3	レステーブル (res)	9
5.4	スレッドカテゴリテーブル (thread_category)	10
5.5	お気に入り掲示板テーブル (favorite_bbs)	10
5.6	コレクトユーザテーブル (collect_user)	11
5.7	不適切な単語登録テーブル (inadequacy_word)	12
5.8	通報テーブル (report)	12
5.9	凍結ユーザテーブル (suspend)	13
5.10	お知らせテーブル (notice)	13
5.11	警告注意テーブル (warn_caution)	14
6	モジュール設計	14
6.1	admin_top.html	14
6.2	admin_child_top.html	15
6.3	create_admin.html	15
6.4	new_admin.html	15
6.5	delete_admin.html	16
6.6	delete_admin_ok.html	16
6.7	user_admin_top.html	16
6.8	create_user.html	16
6.9	new_user.html	17
6.10	search_user.html	17
6.11	user_info.html	17
6.12	.html	18
6.13	login_top.html	18
6.14	login_user.html	18
6.15	subscribe_user.html	19
6.16	subscribe_ok.html	19

6.17	read_news.html	19
6.18	login_admin.html	20
6.19	news_edit.html	20
6.20	news_edit_confirm.html	20
6.21	news_edit_ok.html	21
6.22	search.html	21
6.23	report_reason.html	21
6.24	report_confirm.html	21
6.25	report_ok.html	22
6.26	NGword.html	22
7	ルーティング及び MVC 一覧	23
7.1	ルーティング一覧	24
7.2	ルーティング (管理者側) 一覧	24
7.3	Controller	25
7.3.1	sessios_controller.rb	25
7.3.2	passwords_controller.rb	26
7.3.3	home_controller.rb	26
7.3.4	users_controller.rb	26
7.3.5	results_controller.rb	26
7.3.6	results_categories_controller.rb	27
7.3.7	infomations_controller.rb	27
7.3.8	categories_controller.rb	27
7.3.9	threads_controller.rb	27
7.3.10	contents_controller.rb	27
7.4	Controller 層 (管理者側)(仮)	28
7.4.1	admins_session_controller.rb	28
7.4.2	admins_top_controller.rb	28
7.4.3	ng_word_controller.rb	29
7.4.4	manage_controller.rb	29
7.4.5	manage_user_controller.rb	29
7.4.6	announce_controller.rb	30
7.4.7	report_controller.rb	30

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 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。

5.1 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 2 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	char	20	×	-	-
パスワード	password	char	20	×	-	-
保有ポイント	now_point	int	4	×	-	-
取り消しフラグ	cancel_flag	int	1	×	-	-
色フラグ	color_flag	int	1	×	-	-
斜線フラグ	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。値は 7 文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- ユーザ ID
ユーザを識別するための ID を示す値である。値は最大 20 文字の半角英字と半角数字の組み合わせにて構成される。
例としてユーザ ID「user1234」がある。
- 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。値は最大 20 文字の半角英数字と半角数字の組み合わせにて構成される。
例としてパスワード「pass1234」がある。
- 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。値は 4 桁の数値にて構成される。例として保有ポイント「9999」がある。
- 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。値は 1 桁の 0(OFF)、1(ON) の数値にて構成される。
例として取り消しフラグを 1 にした場合、ユーザの拡張機能「取り消し線」を付与する。
- 色フラグ
拡張機能の 1 つであり、色を付与することを示す値である。値は 1 桁の 0(OFF)、1(ON) の数値にて構成される。
例として色フラグを 1 にした場合、ユーザの拡張機能「赤色」を付与する。

- 斜線フラグ
拡張機能の1つであり、斜線を付与することを示す値である。値は1桁の0(OFF)、1(ON)の数値にて構成される。
例として斜線フラグを1にした場合、ユーザの拡張機能「斜線」を付与する
- 太文字フラグ
拡張機能の1つであり、太文字を寄与することを示す値である。値は1桁の0(OFF)、1(ON)の数値にて構成される。
例として太文字フラグを1にした場合、ユーザの拡張機能「太文字」を付与する。
- 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。
例として管理者フラグを1にした場合、管理者は「親管理者」と識別される。
- 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。値は1桁の0(OFF)、1(ON)の数値にて構成される。
例として初回ログインフラグを1にした場合、「初回ログイン」と識別される。
- ログイン日数
ユーザがログインをした日数の回数を記録することを示す値である。値は4桁の数値にて構築される。
例としてログイン日数「9999」がある。

5.2 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	char	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 updat current_timestamp	-

- スレッド ID
自テーブルの主キーである。値は5桁の数値に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。値は3桁の数値に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。値は7文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- スレッドタイトル名
ユーザが作成したスレッドタイトルを示す値である。値は最大50文字の全角文字にて構成される。
例としてユーザがスレッドタイトル「研究室どこにしようか」がある。
- 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するを示す値である。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、「スレッドを非表示」にする。
- 作成日時
レコードを作成した日付・時刻を示す値である。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。レコードが更新されるたびに自動更新を行う。

5.3 レステーブル (res)

レステー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	5	×	unsigned zerofill auto_increment	PK
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	FK:thread
学籍番号	student_id	char	7	×	-	FK:user
書き込み ID	ress_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	×	-	-
レス非表示フラグ	ress_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。値は5桁の数値で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る。値は5桁の数値に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。値は7文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- 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。値は最大8文字の半角英数か半角数字にて構

成される。

例として書き込み ID 「ero1o23u」がある。

- 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。値は 15 文字以下の全角文字にて構成される。

例として書き込み名「名無しさん」がある。

- 書き込み内容

書き込み内容を示す値である。値は 200 文字以下の全角文字にて構成される。

例として書き込み内容「自分のやりたいことができる場所でええんやで」がある。

- コレクトボタンブッシュ数

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。値は 4 桁の数値にて構成される。

(例としてコレクトボタンブッシュ数「10」がある。

- レス非表示フラグ

管理者の操作権限でレスを非表示するを示す値である。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、「レスを非表示」にする。

- 作成日時

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

5.4 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。値は値は 3 桁の数値に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。値は 15 文字以下の全角文字にて構成される。

例としてカテゴリ名「情報学群」がある。

5.5 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

- 学生番号
ユーザテーブルを参照する際の外部キーである。値は 7 文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- 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。値は 5 桁の数値に構築され自動追加される。

5.6 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	5	×	unsigned zerofill auto_increment	FK:res

- 学生番号
ユーザテーブルを参照する際の外部キーである。値は 7 文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- 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。値は 5 桁の数値にて構築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。値は 5 桁の数値にて構築され自動追加される。

5.7 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。値は 5 桁の数値に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。値は 200 文字以下の全角文字にて構成される。
例として単語名「バカ」がある。

5.8 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	char	20	×	-	FK:user
スレッド ID	thread_id	int	5	×	unsigned zerofill auto_increment	FK:thread
レス ID	res_id	int	5	×	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。値は 5 桁の数値に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。値は最大 20 文字の半角英字と半角数字の組み合わせにて構成される。
例として通報ユーザ ID「user1234」がある。
- 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。値は 5 桁の数値に構築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。値は 5 桁の数値に構築され自動追加される。
- 通報内容
通報内容を示す値である。値は 200 文字以下の全角文字にて構成される。
例として通報内容「このユーザ〇〇は誹謗中傷に反している思ったから」がある。
- 通報日時
レコードを作成した日付・時刻を示す値である。レコードが作成されるたびに自動追加を行う。

5.9 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	×	lunsigned 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。値は 4 桁の数値に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。値は 7 文字の半角数字にて構成される。
例として学籍番号「1200***」がある。

5.10 お知らせテーブル (notice)

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	notice_id	int	5	×	lunsigned zerofill auto_increment	PK
お知らせタイトル	notice_title	varchar	50	×	-	-
お知らせ内容	notice_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。値は 4 桁の数値に構成され自動追加される。
- お知らせタイトル
管理者が作成したお知らせタイトルを示す値である。値は 50 文字以下の全角文字にて構成される。例としてお知らせタイトル「現在確認している不具合について」がある。

- お知らせ内容
お知らせ内容を示す値である。値は 400 文字以下の全角文字にて構成される。
例としてお知らせ内容「KUTBBS 運営事務局です。…」がある。
- 作成日時
レコードを作成した日付・時刻を示す値である。レコードが作成されるたびに自動追加を行う。

5.11 警告注意テーブル (warn_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	notice_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	notice_title	varchar	50	×	-	-
警告注意内容	notice_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。値は 4 桁の数値に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。値は 7 文字の半角数字にて構成される。
例として学籍番号「1200***」がある。
- 警告注意タイトル
管理者が作成した警告注意タイトルを示す値である。値は 50 文字以下の全角文字にて構成される。例としてお知らせタイトル「警告勧告について」がある。
- 警告注意内容
警告注意内容を示す値である。値は 400 文字以下の全角文字にて構成される。
例としてお知らせ内容「KUTBBS 運営事務局です。…」がある。
- 作成日時
レコードを作成した日付・時刻を示す値である。レコードが作成されるたびに自動追加を行う。

6 モジュール設計

6.1 admin_top.html

【名称】

管理者 TOP 画面

【概要】

管理者がログインを済ませると最初に現れる画面である。ここでは管理者に与えられた権限の選択をすること

ができる。

【処理フロー】

- 「掲示板はこちらボタン」を押すと、が呼び出される。
- 「子管理者管理ボタン」を押すと、admin_child_top.html が呼び出される。
- 「ユーザ管理ボタン」を押すと、が呼び出される。
- 「お知らせ編集ボタン」を押すと、が呼び出される。
- 「通報状況確認ボタン」を押すと、が呼び出される。
- 「不適切な単語登録ボタン」を押すと、が呼び出される。

6.2 admin_child_top.html

【名称】

子管理者管理 TOP

【概要】

各子管理者について発行・抹消を行うための画面である。

【処理フロー】

- 「子管理者アカウント発行ボタン」を押すと、create_admin.html が呼び出される。
- 「子管理者抹消ボタン」を押すと、delete_admin.html が呼び出される。
- 「管理者 TOP ボタン」を押すと、admin_top.html が呼び出される。

6.3 create_admin.html

【名称】

子管理者アカウント発行画面

【概要】

子管理者アカウントの発行を行うかどうかを決定する画面である。

【処理フロー】

- 「はいボタン」を押すと、.html.erb で処理が行われる。
- .html.erb で新たなアカウントの発行が完了すると new_admin.html が呼び出される。
- 「いいえボタン」を押すと、admin_child_top.html が呼び出される。

6.4 new_admin.html

【名称】

子管理者アカウント発行完了画面

【概要】

子管理者アカウント発行完了画面が表示される

【処理フロー】

- 「確認ボタン」を押すと、admin_child_top.html が呼び出される。
- 「管理者 TOP ボタン」を押すと、admin_top.html が呼び出される。

6.5 delete_admin.html

【名称】

子管理者アカウント抹消画面

【概要】

特定の子管理者アカウントについて抹消するかどうかを決定する画面である。

【処理フロー】

- 「はいボタン」を押すと、.html.erb で処理が行われる
- .html.erb で子管理者アカウントの抹消が完了すると delete_admin_ok.html が呼び出される。
- 「いいえボタン」を押すと、admin_child_top.html が呼び出される。

6.6 delete_admin_ok.html

【名称】

子管理者アカウント抹消完了画面

【概要】

子管理者アカウント抹消完了画面が表示される

【処理フロー】

- 「戻るボタン」を押すと、admin_child_top.html が呼び出される。

6.7 user_admin_top.html

【名称】

ユーザ管理画面

【概要】

ユーザのアカウント発行とユーザ情報検索を行うことができる画面である。

【処理フロー】

- 「ユーザアカウント発行ボタン」を押すと、create_user.html が呼び出される。
- 「ユーザ情報検索ボタン」を押すと、search_user.html が呼び出される。

6.8 create_user.html

【名称】

ユーザアカウント発行画面

【概要】

ユーザアカウントの発行を行うことができる画面である。学籍番号の開始番号と末尾番号を入力することで範囲を指定し、同時に複数のアカウントの発行を行う。

【処理フロー】

- 「開始番号テキストボックス」に新しく発行するユーザの開始番号を入力する。
- 「末尾番号テキストボックス」に新しく発行するユーザの末尾番号を入力する。
- 「登録ボタン」を押すと、.html.erb で処理が行われる。
- .html.erb でユーザのアカウント発行が完了すると new_user.html が呼び出される。
- 「管理者 TOP ボタン」を押すと、admin_top.html が呼び出される。

6.9 new_user.html

【名称】

ユーザアカウント発行確認画面

【概要】

ユーザアカウント発行確認画面が表示される

【処理フロー】

- 「確認ボタン」を押すと、user_admin_top.html が呼び出される。

6.10 search_user.html

【名称】

ユーザ情報検索画面

【概要】

【処理フロー】

- 「学籍番号検索テキストボックス」にはユーザの情報を入手する際に学籍番号を入力する。
- 「検索ボタン」を押すと、.html.erb で処理が行われる。
- .html.erb で該当するユーザがあれば user_info.html が呼び出される。

6.11 user_info.html

【名称】

ユーザ情報画面

【概要】

ユーザ情報の検索結果を表示する画面である。【処理フロー】

- 「警告・注意喚起ボタン」を押すと、が呼び出される。
- 「アカウント凍結ボタン」を押すと、が呼び出される。
- 「パスワード表示ボタン」を押すと、が呼び出される。
- 「管理者 TOP ボタン」を押すと、admin_top.html が呼び出される。

6.12 .html

【名称】

画面

【概要】

【処理フロー】

- 「ボタン」を押すと、が呼び出される。

6.13 login_top.html

【名称】

ログイン TOP 画面

【概要】

利用者が本システムを利用する際の初期画面である。ここでは、ユーザとしてログイン、管理者としてログインのいずれかを選択することができる。

【処理フロー】

- ログインボタンを押すと login_user.html が呼び出される。
- 管理者はこちらボタンを押すと login_admin.html が呼び出される。

6.14 login_user.html

【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】

- 「ID 入力テキストボックス」にユーザの ID を入力する。
- 「パスワード入力テキストボックス」にユーザのパスワードを入力する。
- 初回のログインでは、ID とパスワードが入力された状態で「ENTER ボタン」を押すと subscribe_user.html が呼び出される。
- 2回目以降のログインでは、ID とパスワードが入力された状態で「ENTER ボタン」を押すと BBS_top.html が呼び出される。
- データベース上に存在しない ID またはパスワードが入力された状態で「ENTER ボタン」を押すと、.html.erb が呼び出される。
- いずれかのテキストボックスに何も入力されていない状態で「ENTER ボタン」を押すと、.html.erb が呼び出される。

6.15 subscribe_user.html

【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】

- 「ID 入力テキストボックス」に変更後のユーザ ID を入力する。
- 「パスワード入力テキストボックス」に変更後のユーザのパスワードを入力する。
- 「パスワード再確認入力テキストボックス」に「パスワード入力テキストボックス」に入力したものと
同じパスワードを入力する。
- 「ID」「パスワード」「パスワード再確認」のテキストボックスに文字列が入力されており、かつ「パス
ワード」と「パスワード再確認」のテキストボックスに入力された文字列が一致した場合において、「登
録ボタン」を押すと subscribe_ok.html が呼び出される。
- いずれかのテキストボックスに何も入力されていない状態で「登録ボタン」を押すと、.html.erb が呼び
出される。
- 「登録ボタン」を押したときに「パスワード」と「パスワード再確認」に入力された文字列が合致しな
かった場合、.html.erb が呼び出される。

6.16 subscribe_ok.html

【名称】

新規登録完了画面

【概要】

新規登録が完了したことを表示する画面である。

【処理フロー】

- 「利用を開始するボタン」を押すと、BBS_top.html が呼び出される。

6.17 read_news.html

【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。

6.18 login_admin.html

【名称】

管理者ログイン画面

【概要】

管理者が 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】

- 「管理者 ID 入力テキストボックス」に管理者の ID を入力する。
- 「管理者パスワード入力テキストボックス」に管理者のパスワードを入力する。
- 入力された ID とパスワードがともにデータベースに登録されたものと合致している場合、「ENTER ボタン」を押すと admin_top.html が呼び出される。
- 入力された ID とパスワードのいずれかがデータベースに登録されているものと合致しない場合、「ENTER ボタン」を押すと、.html.erb が呼び出される。
- ID とパスワードのいずれかが入力されていない場合、「ENTER ボタン」を押すと、.html.erb が呼び出される。

6.19 news_edit.html

【名称】

お知らせ編集画面

【概要】

管理者がユーザにお知らせを通知する内容を編集する画面である。

【処理フロー】

- お知らせタイトル、内容に文字列が入力されている状態で「送信ボタン」を押すと、news_edit_confirm.html が呼び出される。
- お知らせタイトル、内容のいずれかに文字列が入力されていない状態で「送信ボタン」を押すと、.html.erb が呼び出される。
- 「管理者 TOP ボタン」を押すと、admin_top.html が呼び出される。

6.20 news_edit_confirm.html

【名称】

お知らせ内容確認画面

【概要】

お知らせ編集画面で編集したお知らせ内容を確認する画面である。内容を確認し、送信するか否かを選択する。

【処理フロー】

- 「はいボタン」を押すと、.html.erb が呼び出される。また、news_edit_ok.html が呼び出される。
- 「いいえボタン」を押すと、news_edit.html が呼び出される。

6.21 news_edit_ok.html

【名称】

お知らせ作成完了画面

【概要】

お知らせの作成が完了したことを示す画面である。

【処理フロー】

- 「戻るボタン」を押すと admin_top.html が呼び出される。

6.22 search.html

【名称】

検索結果画面

【概要】

利用者が検索を行った際に表示される画面。スレッドタイトルを押すとそのスレッドの閲覧画面が表示される。

【処理フロー】

- 「スレッドタイトル」を押すと、read_thread.html を呼び出す。

6.23 report_reason.html

【名称】

通報画面

【概要】

利用者が通報ボタンを押した際に表示される画面。

【処理フロー】

- 「理由テキストボックス」に通報する理由を入力する。
- 「送信ボタン」を押すと、report_confirm.html を呼び出す。

6.24 report_confirm.html

【名称】

通報確認画面

【概要】

利用者が通報画面にて、「送信ボタン」を押した際に表示される。

【処理フロー】

- 「はいボタン」を押すと、report_ok.html を呼び出します。
- 「いいえボタン」を押すと、report_reason.html を呼び出す。

6.25 report_ok.html

【名称】

通報完了画面

【概要】

利用者が通報確認画面にて、「はいボタン」を押した際に表示される画面。

【処理フロー】

- 「戻るボタン」を押すと category_top.html を呼び出す。

6.26 NGword.html

【名称】

不適切な単語の登録画面

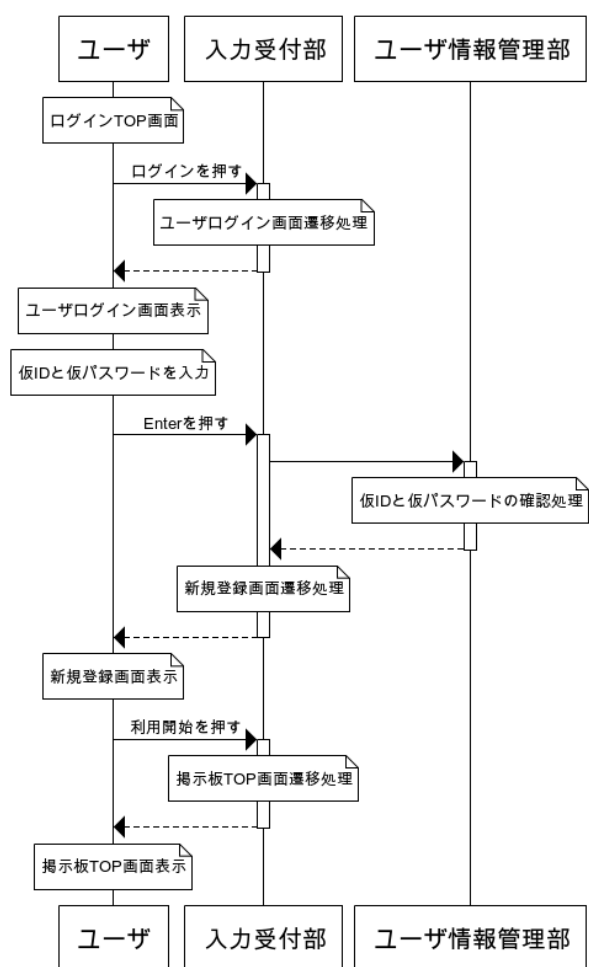
【概要】

管理者が掲示板において不適切な単語を登録する画面。「不適切な単語入力テキストボックス」に入力した単語を「不適切な単語登録ボタン」を押すことで、不適切な単語として登録ができる。表示されている単語の「削除ボタン」を押すことで、その単語を不適切な単語から外す。

【処理フロー】

- 「不適切な単語入力テキストボックス」に単語を入力し、「不適切な単語登録ボタン」押すことで表示されている単語の一覧に入力された単語を追加し、(不適切な単語のデータベース) に追加する。
- 単語の「削除ボタン」を押すことで、不適切な単語の一覧からその単語を外し、(不適切な単語のデータベース) から外す。

アカウント登録登録サブシステム



www.websequencediagrams.com

7 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。

7.1 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

7.2 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

7.3 Controller

7.3.1 sessios_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。

7.3.2 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。

7.3.3 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。検索窓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、/informations/: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 にゲット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。

7.3.4 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。

7.3.5 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.rb 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。

7.3.6 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。

7.3.7 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。

7.3.8 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、検索窓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。

7.3.9 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、/thread/create

-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。

7.3.10 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: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。

7.4 Controller 層 (管理者側)(仮)

7.4.1 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。

7.4.2 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。

7.4.3 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。

7.4.4 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。

7.4.5 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。

7.4.6 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。

7.4.7 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。