

1 モジュール設計

1.1 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。

1.2 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。

1.3 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。

1.4 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。

1.5 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。

1.6 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。

1.7 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。

1.8 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。

1.9 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。

1.10 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。

1.11 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。

1.12 .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。

1.13 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。

1.14 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。

1.15 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一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。

1.16 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。

1.17 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。

1.18 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。

1.19 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。

1.20 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。

1.21 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。

1.22 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。

1.23 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。

1.24 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。

1.25 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。

1.26 NGword.html

【名称】

不適切な単語の登録画面

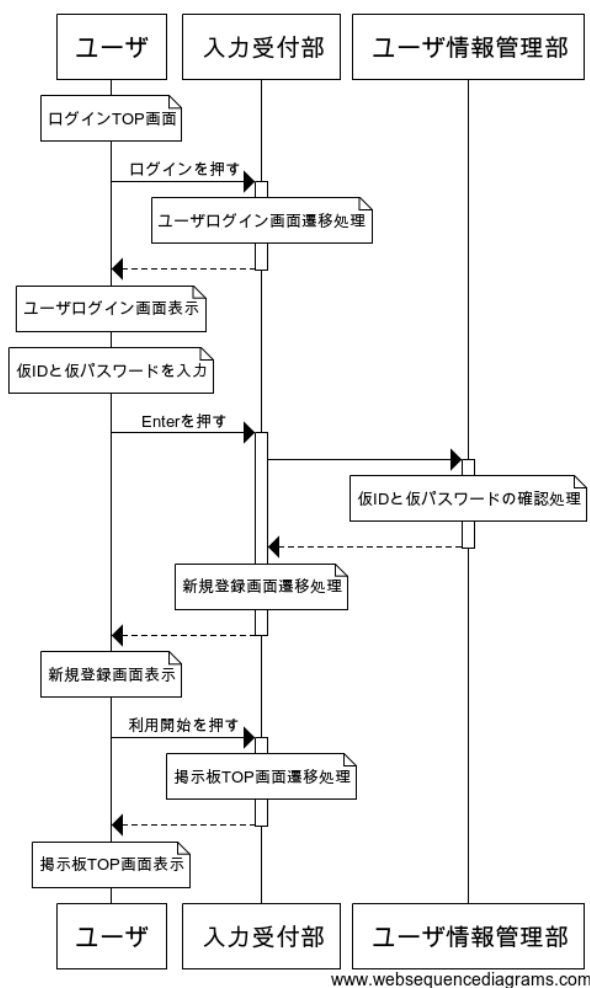
【概要】

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



2 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。

2.1 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 1: 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

2.2 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 2: 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

2.3 Controller

2.3.1 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。

2.3.2 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。

2.3.3 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。

2.3.4 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。

2.3.5 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。

2.3.6 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。

2.3.7 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。

2.3.8 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。

2.3.9 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。

2.3.10 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。

2.4 Controller 層 (管理者側)(仮)

2.4.1 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。

2.4.2 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。

2.4.3 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。

2.4.4 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。

2.4.5 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。

2.4.6 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。

2.4.7 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。