第 1 章

システム概要

1.1　ユーザ側システム一覧

- 基本機能
  - ユーザ登録機能
  - ログイン機能
  - ログアウト機能
  - ユーザ情報編集機能
  - ユーザ削除機能
- 楽譜データに関する機能
  - 楽譜データ登録機能
  - 楽譜データ編集機能
  - 楽譜データ削除機能
  - 楽譜データ一覧閲覧機能
  - 楽譜データ詳細閲覧機能
- 楽譜検索機能
  - 作曲者，編曲者からの検索機能
  - 曲名からの検索機能
  - グレード（難易度）からの検索機能
  - 使用楽器からの検索機能
- 楽譜ソート機能
  - 曲名から，五十音順にソートする機能
  - 作曲者，編曲者を五十音順にソートする機能
  - グレード順にソートする機能
  - 演奏時間の短い順にソートする機能
- お問い合わせ画面表示機能
  - お問い合わせ機能
- 広告表示機能
  - 広告表示機能

1.2　管理者側システム一覧

- 基本機能
  - ログイン機能
  - ログアウト機能
  - 管理者情報編集機能
- ユーザに対する機能
  - ユーザ情報編集機能
  - ユーザ削除機能
  - 楽譜データ削除機能
- ユーザ検索機能
  - ユーザ名から検索する機能
  - ユーザ ID から検索する機能
- ユーザソート機能
  - ユーザ名を五十音順にソートする機能
- 楽譜検索機能
  - 作曲者，編曲者からの検索機能
  - 曲名からの検索機能
  - グレード（難易度）からの検索機能
  - 使用楽器からの検索機能
- 楽譜ソート機能
  - 曲名から，五十音順にソートする機能
  - 作曲者，編曲者を五十音順にソートする機能
  - グレード順にソートする機能
  - 演奏時間の短い順にソートする機能
- 広告登録機能
  - 広告登録機能

第 2 章

モジュール設計

2.1　前提

我々は Ruby on rails を用いて開発を行う．この言語の慣習に則り，これ以降，

- index
- new
- show
- edit
- create
- destroy

の 6 つは「アクション」と呼び，これ以外の関数を全て「メソッド」と呼ぶ.

アクションはコントローラが異なれば同名のものを使用するため，モジュール ID を「Controller 名. アクション名」とする．また，1 アクション 1 機能を持つ.

2.2　モジュール詳細

以下に，各モジュールについて「定義書」「フロー図」の順で示す.

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/1
	作成者	中村祐貴	更新日	2023/12/6
基本情報	概要	ScoresController 内の show アクション		
	所属クラス	ScoresController		
	モジュール名	show		
	モジュールID	ScoresController.show		

処理説明

【処理内容】

楽譜の詳細画面を表示する．

【処理手順】

1. Score クラスの find メソッドを用いて，インスタンスを変数 @score に格納する．
   find メソッドの引数は params[:id] とする．
2. views/scores/show.html.erb を描画する．

【補足】

入力値説明

score_id

出力値説明

なし

他クラス・関数との関係

ApplicationController を継承する．

図 2.1: ScoresController.show 定義書

所属クラス	ScoresController	作成日	2023/12/7
機能名	楽譜情報詳細	更新日	2023/12/8
モジュールID	ScoresController.show	作成者	中村祐貴
使用モジュールID			

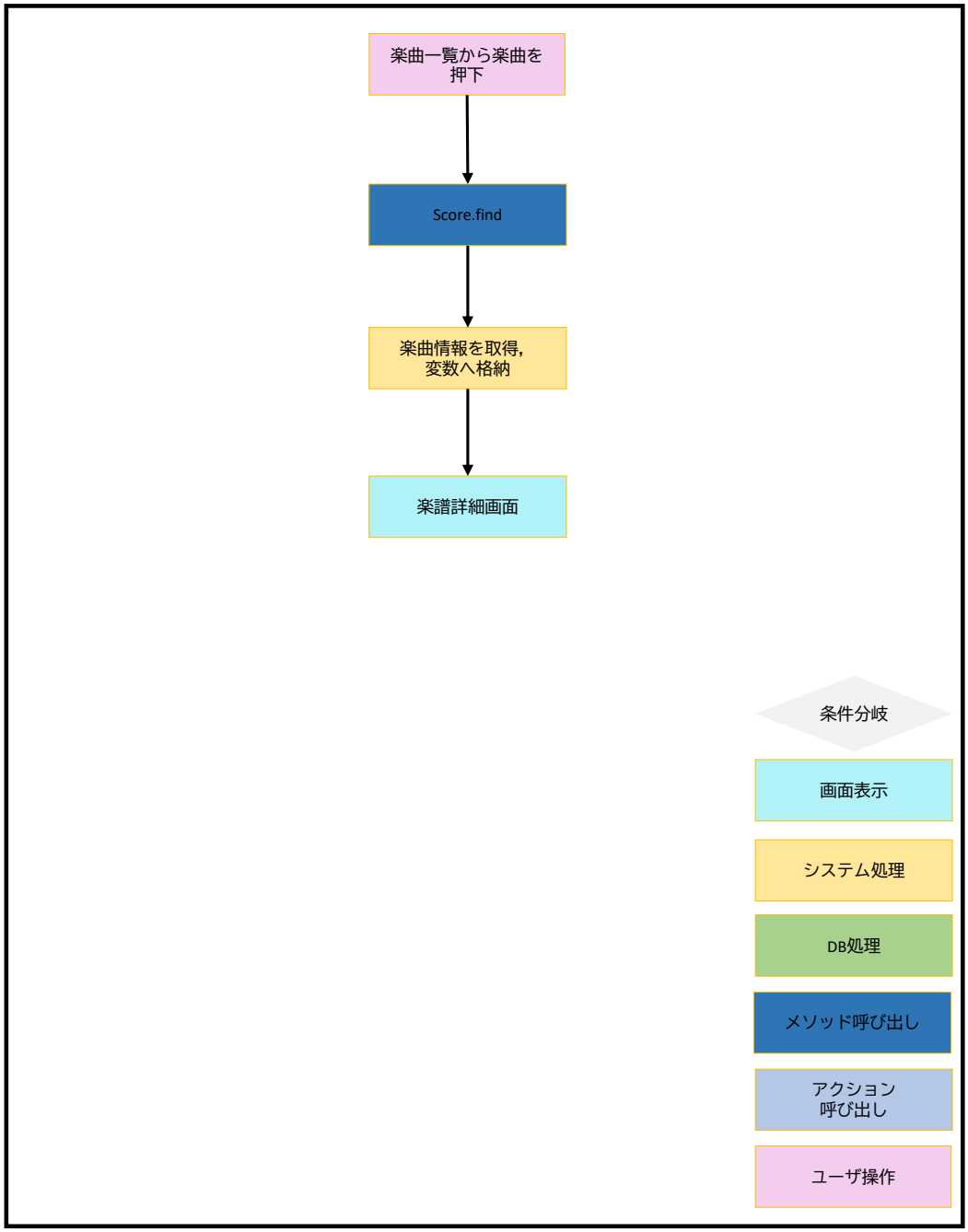

図 2.2: ScoresController.show フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/1
	作成者	中村祐貴	更新日	2023/12/7
基本情報	概要	UsersController 内の show アクション		
	所属クラス	UsersController		
	モジュール名	show		
	モジュールID	UsersController.show		

処理説明

【処理内容】
ユーザの詳細画面を表示する．
【処理手順】
1. current_user が admin か user か判定する．current_user が uesr なら自身の user_id しか取得できない．
（admin ならユーザの詳細画面，user なら自身のユーザ情報詳細画面が表示される．）
2. User クラスの find メソッドを用いて，インスタンスを変数 @user に格納する．
find メソッドの引数は params[:id] とする．
3. views/users/show.html.erb を描画する．
【補足】
ユーザが url 直接入力により，他のユーザ情報を閲覧しようとすると自身のユーザ情報画面が表示される．

入力値説明

user_id

出力値説明

なし

他クラス・関数との関係

ApplicationController を継承する．

図 2.3: UsersController.show 定義書

所属クラス	UsersController	作成日	2023/12/7
機能名	ユーザ情報詳細	更新日	2023/12/8
モジュールID	UsersController.show	作成者	中村祐貴
使用モジュールID			

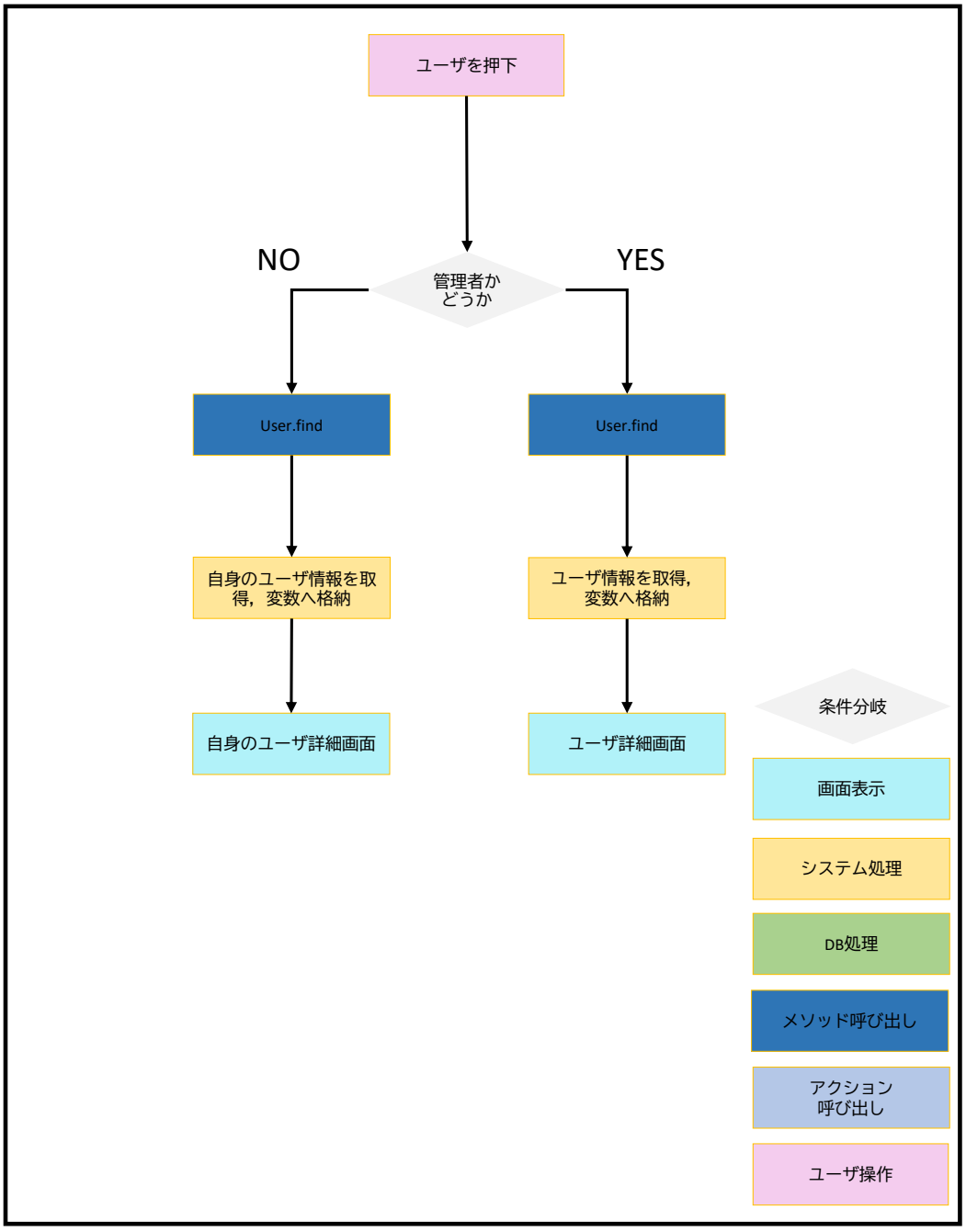

図 2.4: UsersController.show フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/4
	作成者	山本祥弘	更新日	2023/12/5
基本情報	概要	ScoresController 内の new アクション		
	所属クラス	ScoresController		
	モジュール名	new		
	モジュールID	ScoresController.new		

処理説明

【処理内容】

新しい楽譜データを作成するページを表示するアクション.

【処理手順】

1. views/scores/new.html.erb を描画する.

【補足】

入力値説明

なし

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する.

図 2.5: ScoresController.new 定義書

所属クラス	ScoresController	作成日	2023/12/07
機能名	ユーザ削除	更新日	2023/12/07
モジュールID	ScoresController.new	作成者	山本祥弘
使用モジュールID			

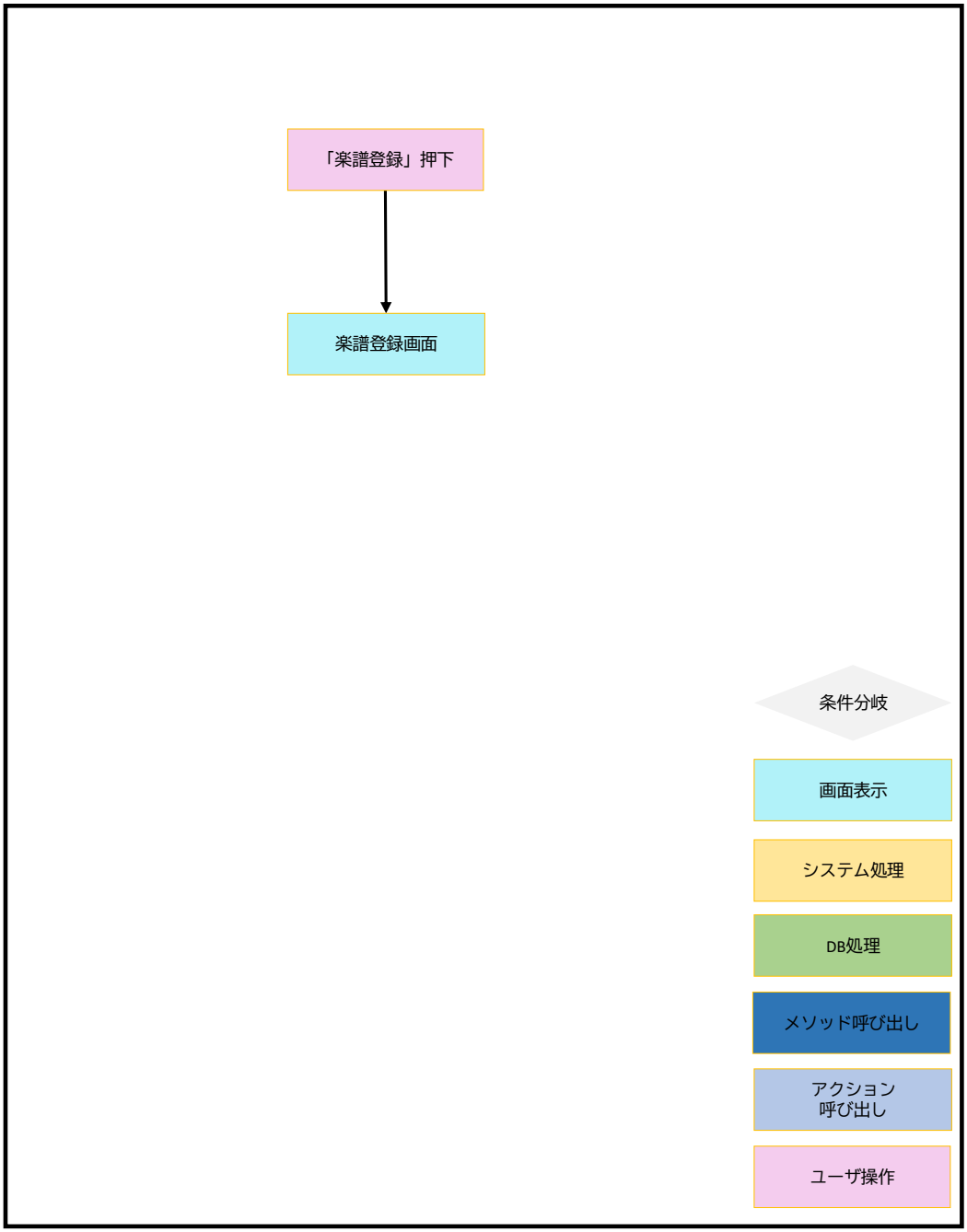

図 2.6: ScoresController.new フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/4
	作成者	山本祥弘	更新日	2023/12/5
基本情報	概要	UsersController 内の new アクション		
	所属クラス	UsersController		
	モジュール名	new		
	モジュールID	UsersController.new		

処理説明

【処理内容】

サインアップ画面を表示するアクション.

【処理手順】

1. views/users/new.html.erb を描画する.

【補足】

入力値説明

なし

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する.

図 2.7: UsersController.new 定義書

所属クラス	UsersController	作成日	2023/12/07
機能名	サインアップ画面表示	更新日	2023/12/08
モジュールID	UsersController.new	作成者	山本祥弘
使用モジュールID			

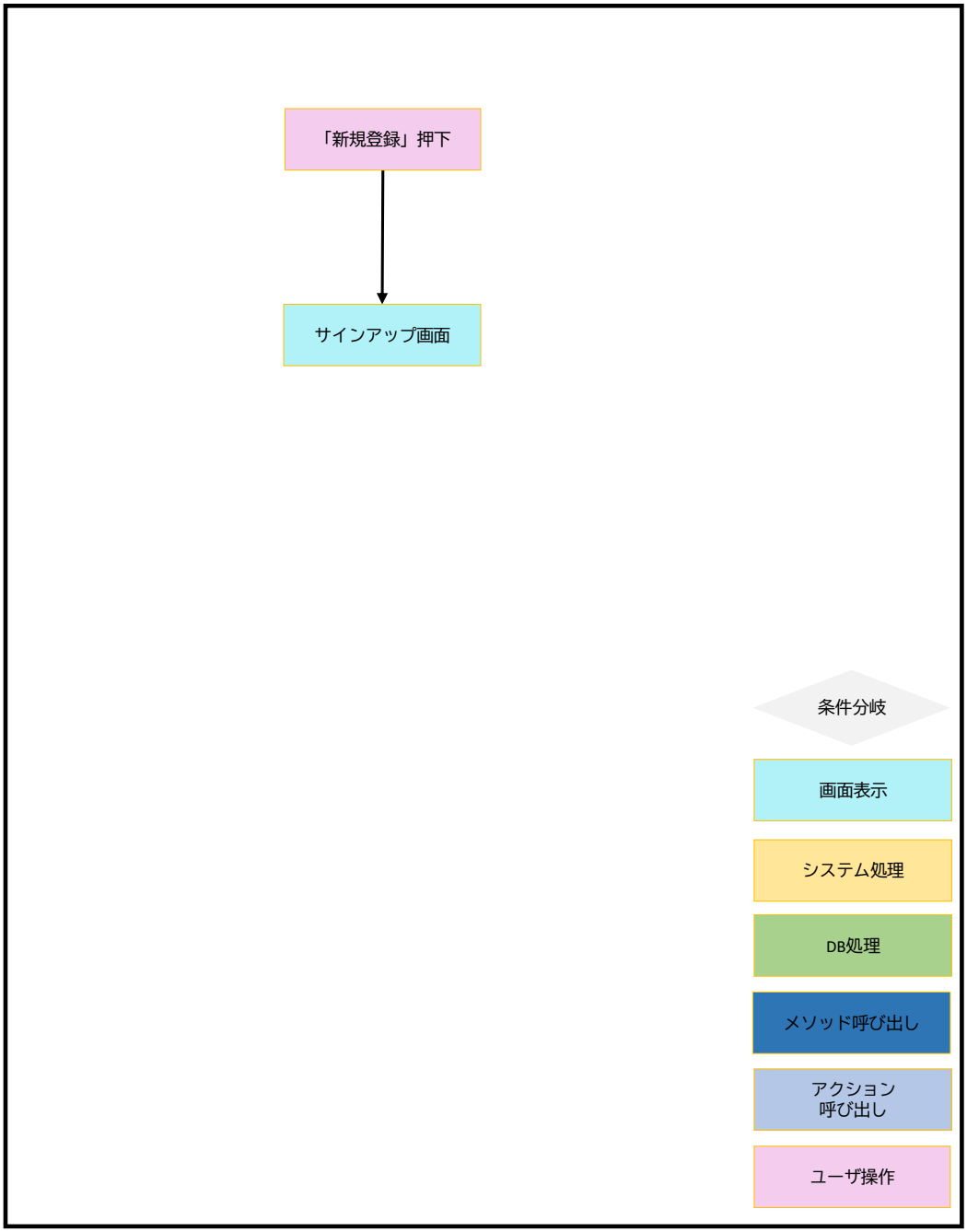

図 2.8: UsersController.new フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/3
	作成者	山田滉希	更新日	2023/12/5
基本情報	概要	ScoresController 内の edit アクション		
	所属クラス	ScoresController		
	モジュール名	edit		
	モジュールID	ScoresController.edit		

処理説明

【処理内容】

楽譜データについて編集ページを作成する．

【処理手順】

1. current_user_id と編集する対象の user_id が一致しているかの確認を行う．
2. 一致しておれば Score クラスの find メソッドを用いて，インスタンスを変数 @score に格納する．（find メソッドの引数は params[:id] とする．）
3. views/scores/edit.html.erb を描画する．

【補足】

入力値説明

score_id，user_id，current_user_id

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する．

図 2.9: ScoresController.edit 定義書

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/3
	作成者	山田滉希	更新日	2023/12/5
基本情報	概要	UsersController 内の edit アクション		
	所属クラス	UsersController		
	モジュール名	edit		
	モジュールID	UsersController.edit		

処理説明

【処理内容】

ユーザについての編集ページを作成する．

【処理手順】

1. current_user_id と admin_id が一致しているか判定し， 一致しなければ変数 @user には current_user_id に紐づけられた user_id しか格納できない．

2. 一致しているなら変数 @user には任意の user_id を格納できる．

2. User クラスの find メソッドを用いて，インスタンスを変数 @user に格納する．(find メソッドの引数は params[:id] とする．)

3. views/users/edit.html.erb を描画する．

【補足】

入力値説明

user_id，current_user_id

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する．

図 2.10: UsersController.edit 定義書

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	1.0.2
	工程名	内部設計	作成日	2023/12/03
	作成者	三上柊	更新日	2023/12/08
基本情報	概要	ScoresController 内の create アクション		
	所属クラス	ScoresController		
	モジュール名	create		
	モジュールID	ScoresController.create		

処理説明

【処理内容】
新たに作成された楽譜データをDBに登録する．

【処理手順】
1. score_params メソッドにて入力を取得する
2. Score クラスの new メソッドに score_params の戻り値を引数として渡し， @score 変数に結果を格納する．
3. save メソッドを用いてDBに @score の中身を保存する．
4. 保存が成功しているかを判定する．
5. 成功していれば成功を意味する flash を含めて ScoresController.index を呼び出す．
6. 失敗していればエラーを意味する flash を含めて ScoresController.new を呼び出す．

【補足】
・score_params メソッドは private である．new での入力内容を取得する．
・flash とは，リダイレクト時に，ページ上に一度だけメッセージを表示するメソッドである．
・saveとは，Railsに標準搭載されたDBにインスタンスを追加するメソッドである．

入力値説明

:composer：作曲者，:arranger：編曲者，
:name：曲名，:grade：難易度，:time：演奏時間，
使用楽器：:piccolo：ピッコロ, :c_flute:フルート 等

出力値説明

他クラス・関数との関係

ApplicationController クラスを継承する．
score_params メソッドを作成，利用する．このメソッドは private である．

図 2.11: ScoresController.create 定義書

所属クラス	ScoresController	作成日	2023/12/06
機能名	楽曲登録	更新日	2023/12/08
モジュールID	ScoresController.create	作成者	三上柊
使用モジュールID	ScoresController.index， UsersController.home		

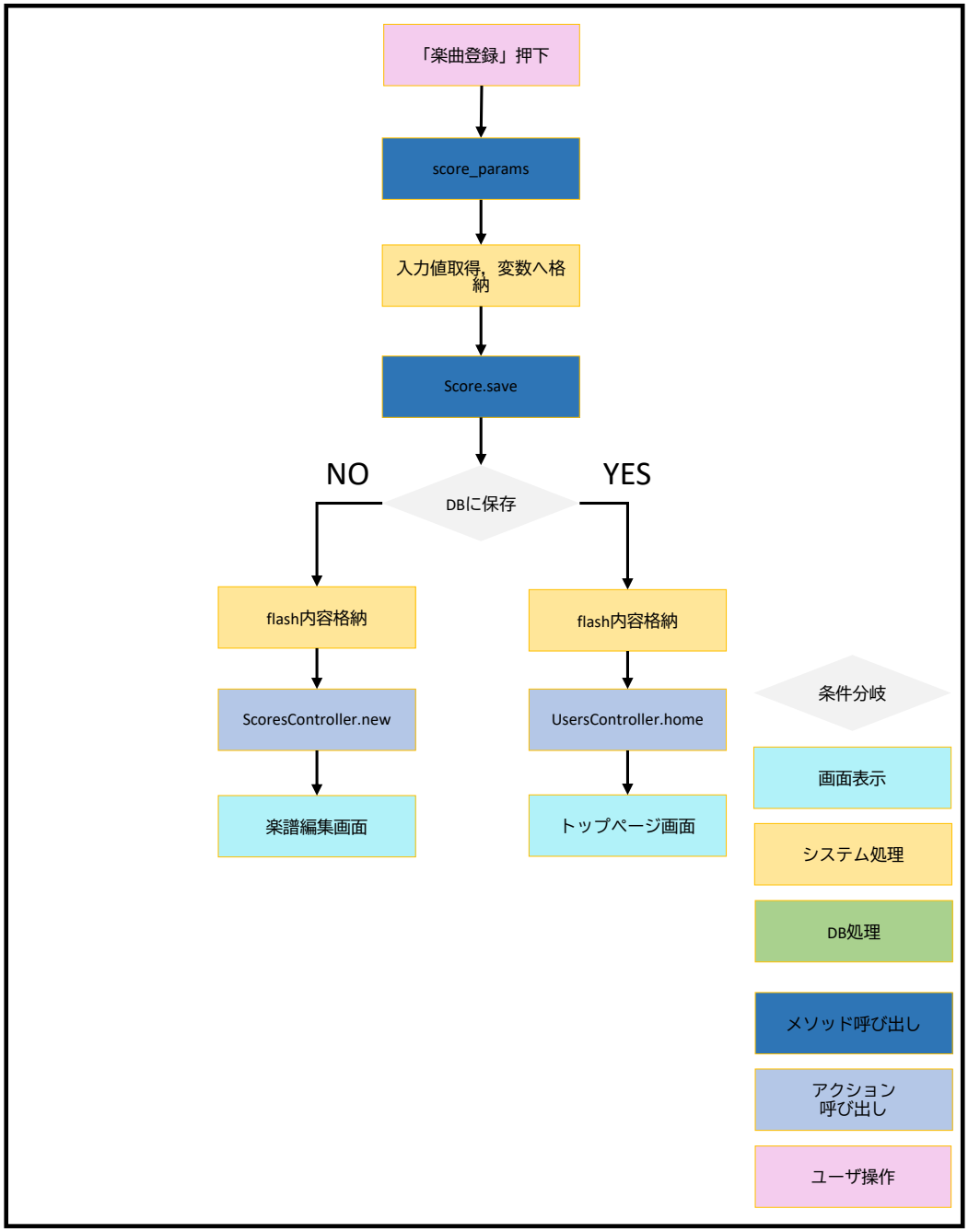

図 2.12: ScoresController.create フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	1.0.2
	工程名	内部設計	作成日	2023/12/03
	作成者	三上柊	更新日	2023/12/08
基本情報	概要	UsersController 内の create アクション		
	所属クラス	UsersController		
	モジュール名	create		
	モジュールID	UsersController.create		

処理説明

【処理内容】
新規のユーザをDBに登録する．
【処理手順】
1. user_params メソッドを用いて new ページでの入力内容を取得する．
2. User クラスの new メソッドを用いて，入力をインスタンス変数 @user に格納する．
3. save メソッドを用いて @user をDBに保存する．
4. 保存が成功したか判定する．
5. 成功していればログイン処理を行い，成功を意味する flash を含めて，ScoresController.index を呼び出す．
6. 失敗していたらエラーを意味する flash を含めて UsersController.new を呼び出す．
【補足】
・このアクションは new ページにて「登録」が押された際に呼び出される．
・「ログイン処理」は SessionsController.create を参照．
・user_params メソッドは private である．new での入力内容を取得する．
・flash とは，リダイレクト時に，一度だけページ上にメッセージを表示させるメソッドである．
・saveとは，Railsに標準搭載された，DBにインスタンスを追加するメソッドである．

入力値説明

:name：ユーザーネーム, :email：メールアドレス,
:pasword：パスワード, :password_confirmation：パスワード再確認

出力値説明

他クラス・関数との関係

ApplicationController クラスを継承する．
user_params メソッドを使用する．

図 2.13: UsersController.create 定義書

所属クラス	UsrsController	作成日	2023/12/06
機能名	ユーザ新規登録	更新日	2023/12/08
モジュールID	UsersController.create	作成者	三上柊
使用モジュールID	UsersController.new， UsersContoroller.home		

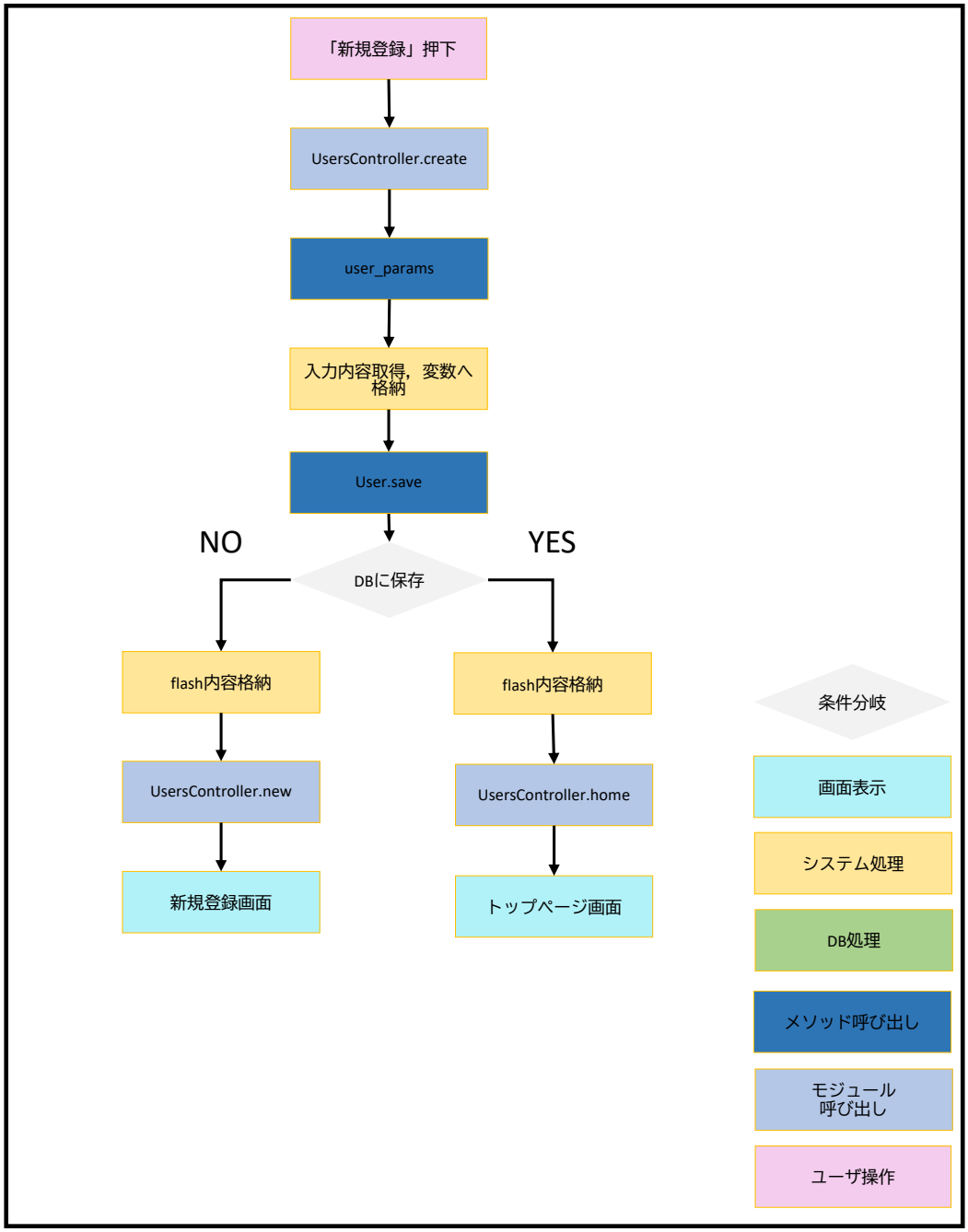

図 2.14: UsersController.create フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/4
	作成者	山本祥弘	更新日	2023/12/8
基本情報	概要	ScoresController 内の destroy アクション		
	所属クラス	ScoresController		
	モジュール名	destroy		
	モジュールID	ScoresController.destroy		

処理説明

【処理内容】
楽譜削除ボタンが押されたときに呼び出す．楽譜データを削除する．
【処理手順】
1. current_user.id が管理者もしくはログインしたユーザであるかを判定する．
2. current_user.id がログインしたユーザである場合，ユーザ削除確認ボックスを表示する．
current_user.id が管理者であった場合，パスワード認証を含むユーザ削除ボックスを表示する．
3.「戻る」ボタンが押されると UsersController.home を呼び出し処理を終了する．「削除」ボタンが押されると， Score.find(id).destroy を実行し，楽譜データを削除する．楽譜データを削除する際,
current_user.id が管理者である場合にはパスワード認証を通過している必要がある．
4. UsersController.home を呼び出し，削除完了の flash を表示する．
【補足】

入力値説明

:score_id : スコアID，:user_id : ユーザID

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する．

図 2.15: ScoresController.destroy 定義書

所属クラス	ScoresController	作成日	2023/12/07
機能名	楽譜データ削除	更新日	2023/12/08
モジュールID	ScoresController.destroy	作成者	山本祥弘
使用モジュールID	UsersController.home		

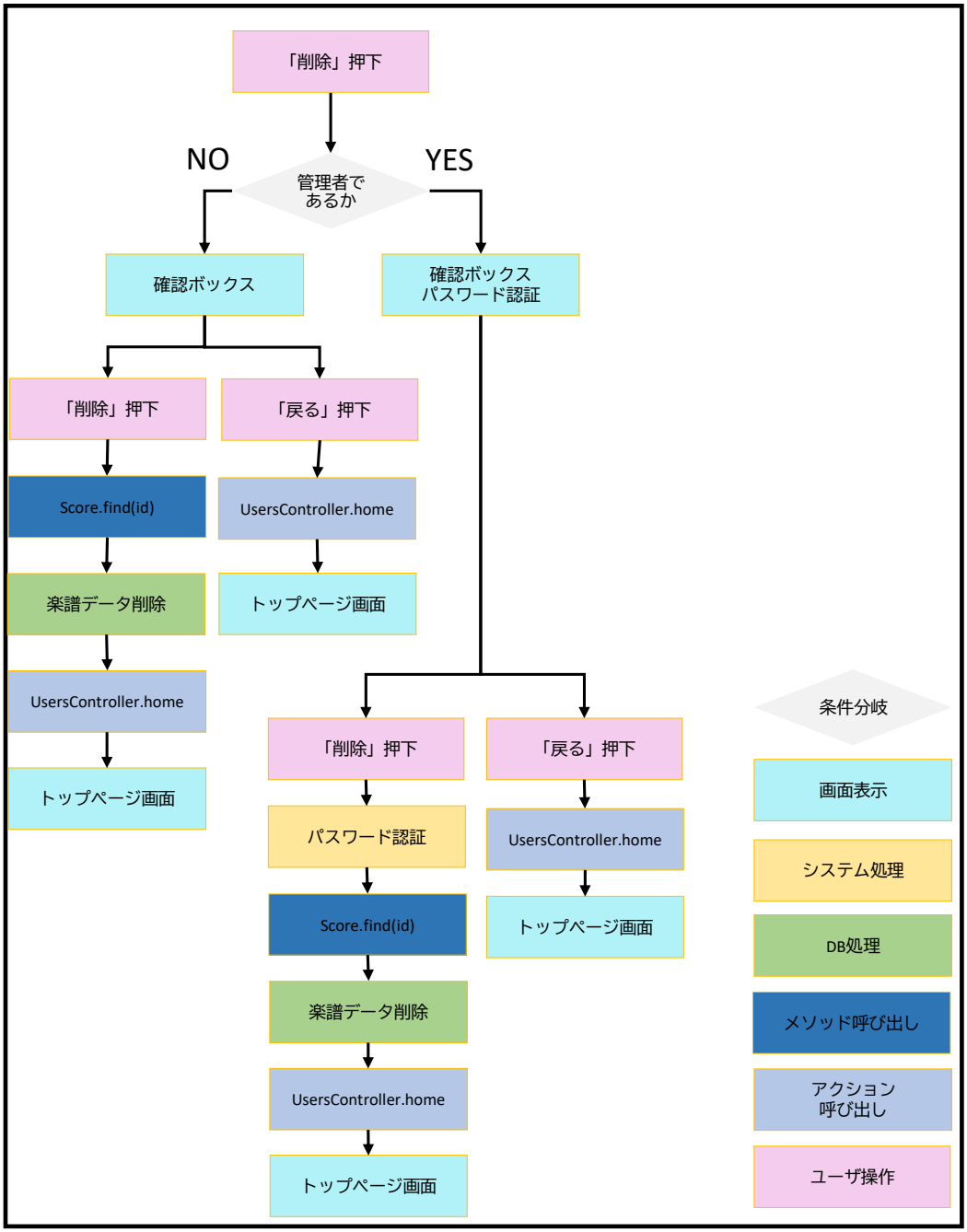

図 2.16: ScoresController.destroy フロー図

モジュール定義				
管理情報	システム名	楽譜管理ソフトウェア	バージョン	
	工程名	内部設計	作成日	2023/12/4
	作成者	山本祥弘	更新日	2023/12/5
基本情報	概要	UsersController 内の destroy アクション		
	所属クラス	UsersController		
	モジュール名	destroy		
	モジュールID	UsersController.destroy		

処理説明

【処理内容】
ユーザ削除ボタンが押されたときに呼び出す．データベースに登録されているユーザを削除する．
【処理手順】
1. current_user.id が管理者もしくはログインしたユーザであるかを判定する．
2. current_user.id がログインしたユーザである場合，ユーザ削除確認ボックスを表示して手順3-1に移行する．current_user.id が管理者であった場合，パスワード認証を含むユーザ削除ボックスを表示して手順4-1に移行する．
3-1.「戻る」ボタンが押されると UsersController.show を呼び出し処理を終了する．「削除」ボタンが押されると， current_user.id と削除するユーザが同じであるかを判定し，同じである場合のみ User.find(id).destroy を実行してユーザを削除する．
3-2. UsersController.new を呼び出し，削除完了の flash を表示して処理を終了する．
4-1.「戻る」ボタンが押されると UsersController.index を呼び出し処理を終了する．パスワード認証を通過し，かつ「削除」ボタンが押されると， User.find(id).destroy を実行し，ユーザを削除する．
4-2. UsersController.index を呼び出し，削除完了の flash を表示する．
【補足】
命令 has_many :scores, dependent: :destroy により，削除されるユーザに関連づけられた楽譜データも同時に削除される．

入力値説明

:user_id : ユーザID

出力値説明

なし

他クラス・関数との関係

ApplicationController クラスを継承する．

図 2.17: UsersController.destroy 定義書

所属クラス	UsersController	作成日	2023/12/07
機能名	ユーザ削除	更新日	2023/12/07
モジュールID	UsersController.destroy	作成者	山本祥弘
使用モジュールID	UsersController.show， UsersController.new， UsersController.index		

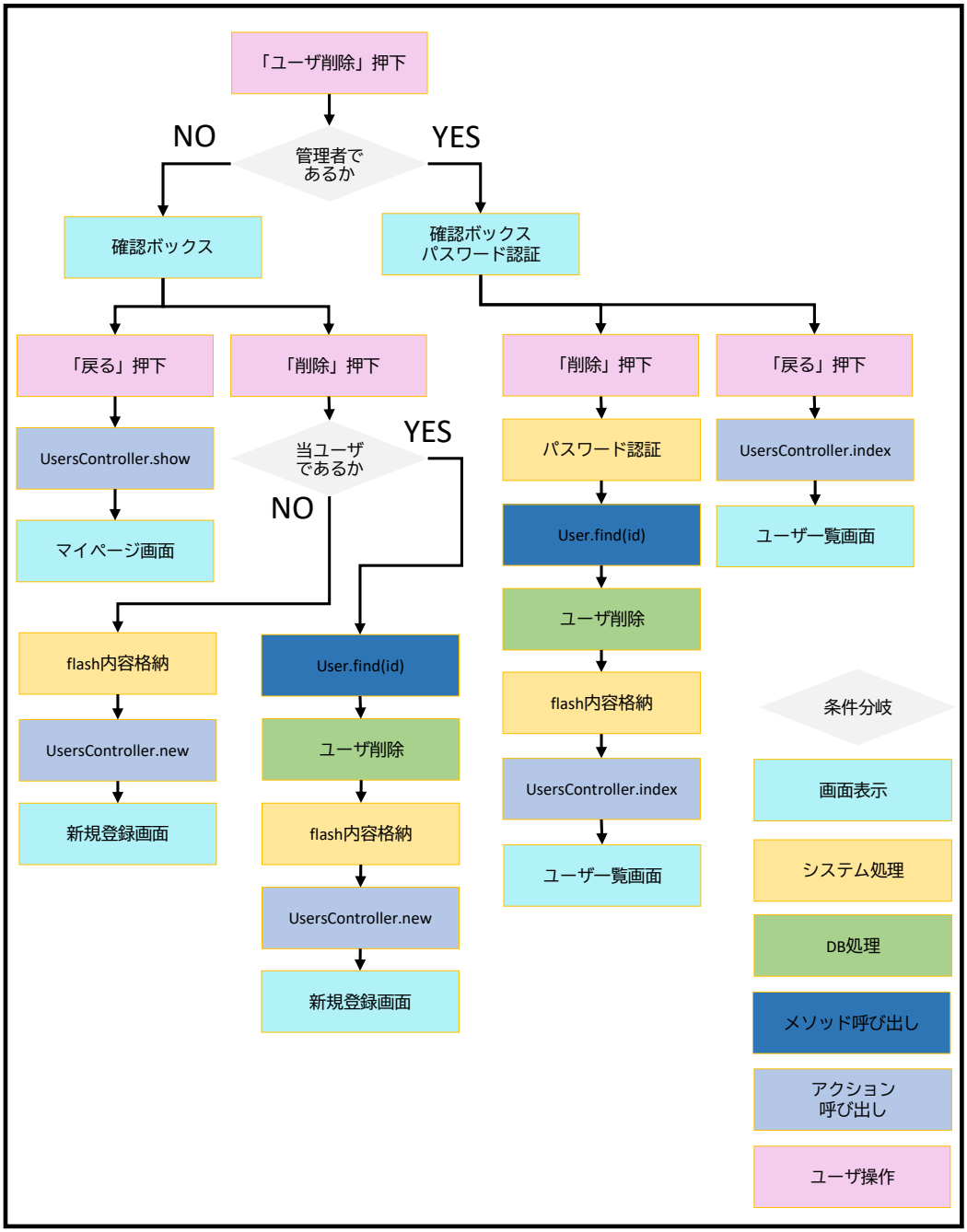

図 2.18: UsersController.destroy フロー図

第 3 章

貢献内容

システム提案書第 2 版の作成における，各メンバの貢献内容は以下の通り．

学籍番号	氏名	貢献内容・担当箇所
1250297	奥平 舜理	モジュールの内容，システム実装方法 モジュール担当箇所：texttt`UC.index`，`UC.home`
1250341	田中 諒	モジュールの内容，テンプレート作成 モジュール担当箇所：texttt`UC.update`，`SC.update`
1250352	中村 祐貴	モジュールの作成，規約（TEX） モジュール担当箇所：texttt`UC.show`，`SC.show`
1250372	三上 柊	モジュールの内容，テンプレート作成・更新，進捗管理，TEX 清書 モジュール担当箇所：texttt`UC.create`，`SC.create`
1250373	溝口 洸熙	モジュールの内容，規約 モジュール担当箇所：texttt`SsC.create`，`SsC.new`，`Ssc.destroy`
1250382	山田 滉希	モジュールの内容 モジュール担当箇所：texttt`UC.edit`，`SC.edit`
1250385	山本 祥弘	モジュールの内容，テンプレート更新 モジュール担当箇所：texttt`UC.destroy`，`SC.destroy`，`UC.destroy`，`SC.destroy`

- `UC`：`UsersController`
- `SC`：`ScoresController`
- `SsC`：`SessionsController`