

# Webアプリケーション 開発実習

共有型TODO管理アプリケーション

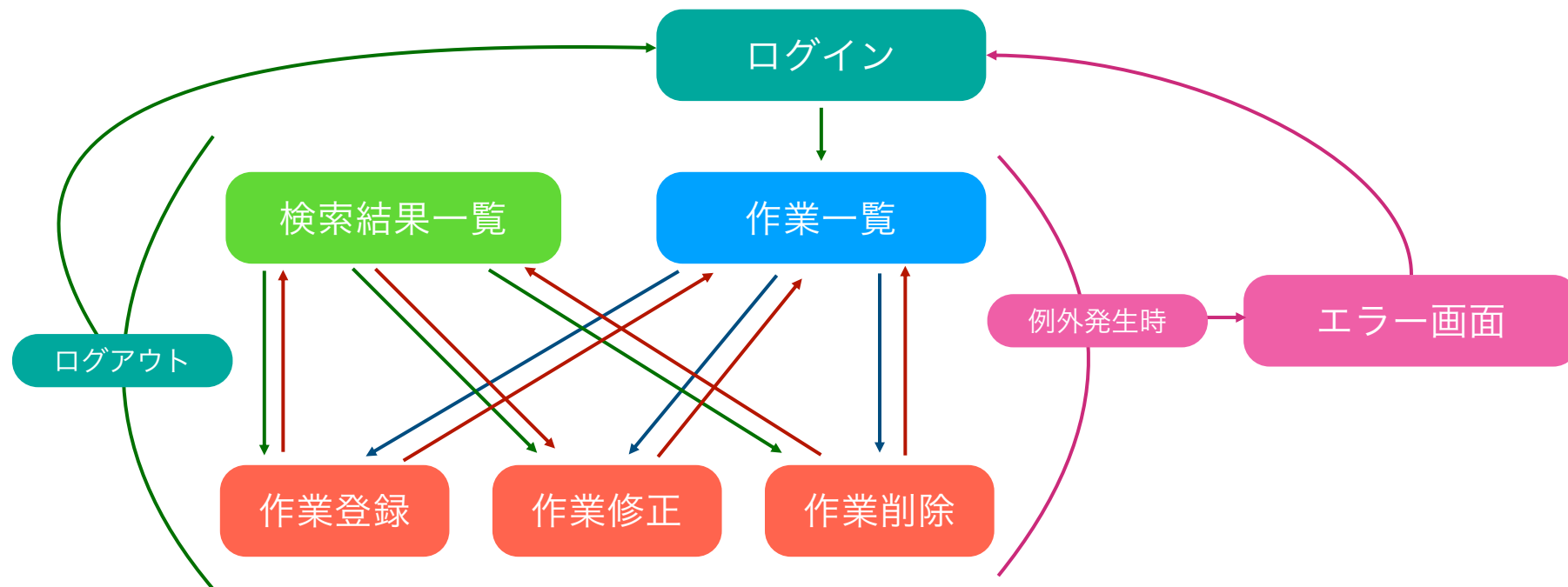
# 共有型TODO管理アプリケーションの仕様

- データはデータベースに登録
  - MySQLを利用
- ログイン、ログアウト機能
- 作業管理機能
  - 作業一覧の表示・検索
  - 作業の追加
  - 作業の完了・未完了の更新
  - 作業の情報（項目名・担当者など）の更新
  - 作業の削除
- ユーザー管理機能
  - ユーザー管理機能は省略します。（SQLでデータベースに登録します）

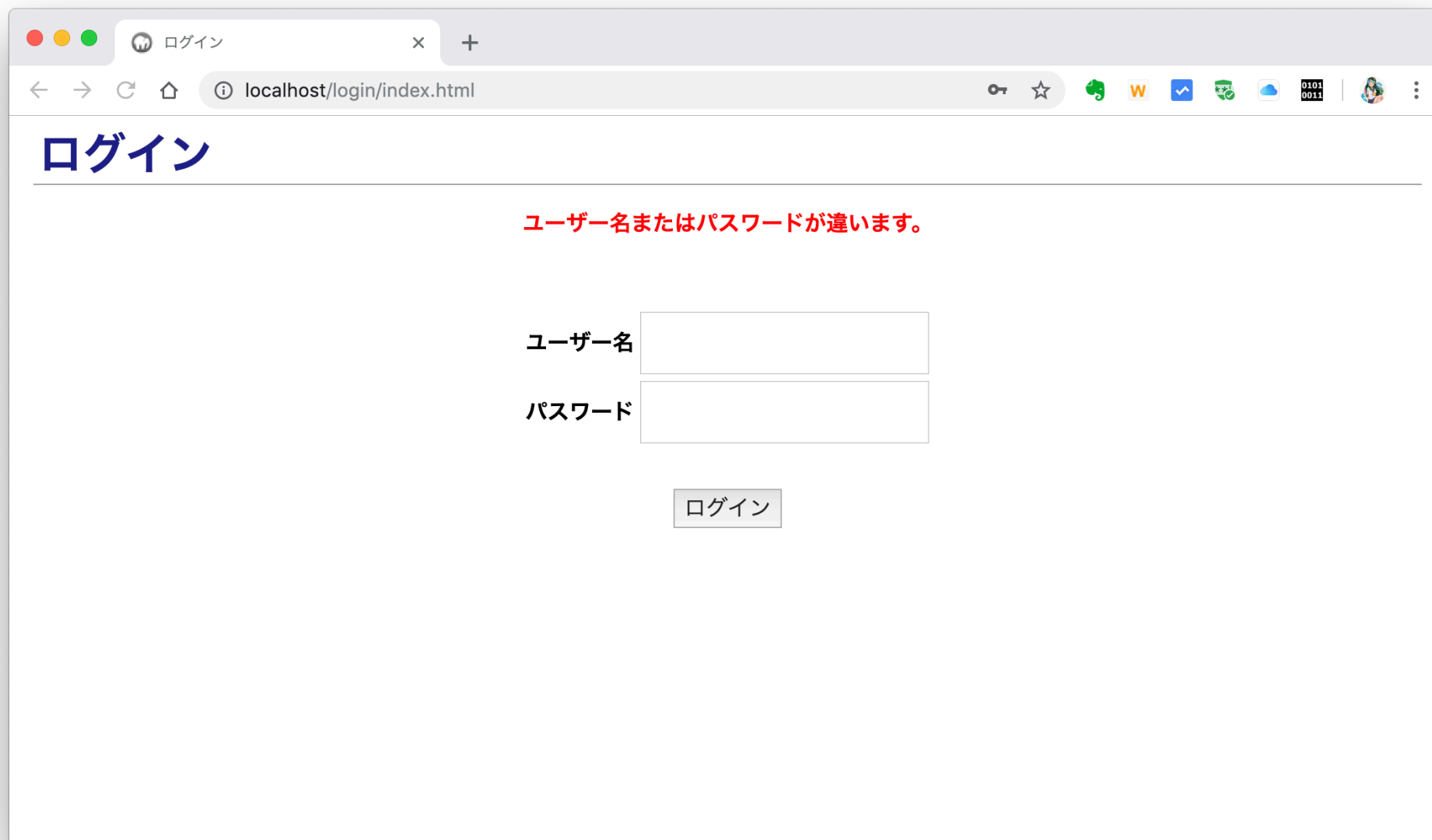
# 画面フロー（画面共通仕様）

- ログイン
  - 未ログイン時はログインページを表示します。
  - 正常にログインできなかったときは、再度ログインページを表示し、「ユーザー名またはパスワードが違います。」と表示します。
  - 正常にログインできた時は、登録された作業一覧ページを表示します。
- ログアウト
  - ログインページ以外の各ページに「ログアウト」ボタンを表示します。
  - ログアウトボタンをクリックするとログアウトし、ログインページを表示します。
- エラー表示について
  - 入力値のバリデーション（値の妥当性チェック）のエラーは、入力画面の上部にエラー内容を表示します。
  - 例外発生時のエラーは、共通のエラーページを表示し、ログインページへのリンクボタンを設置します。
- ログインページ、エラーページ以外のページ
  - ページの右上にログインユーザーの姓名を表示します。

# 画面フロー



# UI設計（画面設計）ログイン



The image shows a web browser window with a single tab titled "ログイン". The address bar displays "localhost/login/index.html". The page content includes a large blue heading "ログイン", a red error message "ユーザー名またはパスワードが違います。", and two input fields labeled "ユーザー名" and "パスワード". A "ログイン" button is positioned below the fields.

ログイン

ユーザー名またはパスワードが違います。

ユーザー名

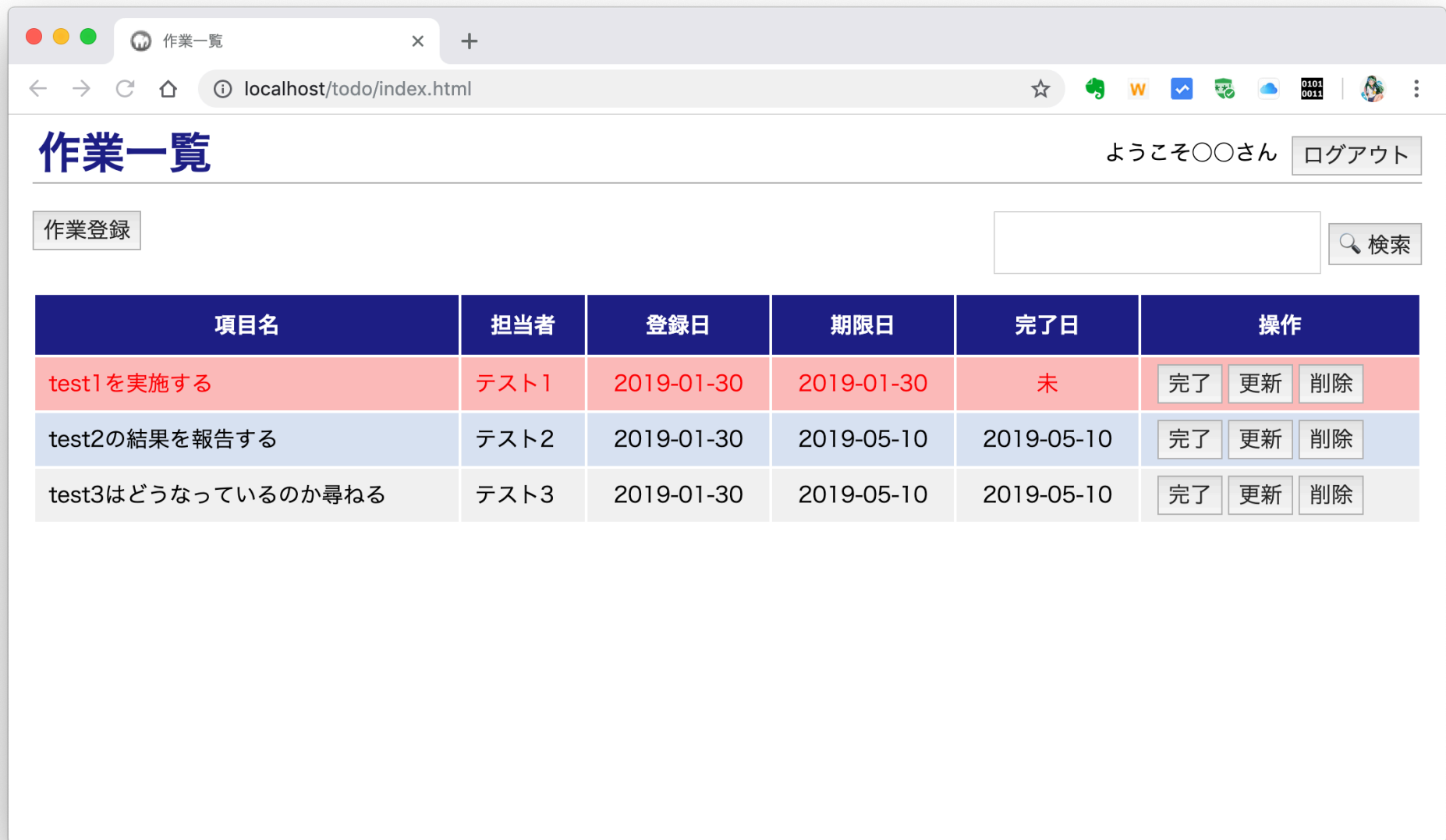
パスワード

ログイン

# UI設計（画面設計） ログイン

- ・ ログイン
  - ・ ユーザー名、パスワードを入力して、「ログインボタン」を押下します。
- ・ データベースを検索します
  - ・ ユーザー名、パスワードに一致するユーザーがあればログイン処理を行い、「作業一覧ページ」を表示します。
  - ・ ユーザー名、パスワードに一致するユーザーがなければ、「ログインページ」を表示し、「ユーザー名またはパスワードが違います。」と表示します。
  - ・ 削除済みのユーザーはログインできません。
- ・ 未ログインの場合
  - ・ 必ずログインページを表示するようにします。

# UI設計（画面設計）作業一覧



# UI設計（画面設計）作業一覧

- 一覧表示
  - 未削除の作業項目を一覧表示します。
  - 表示の順番は、期限日の古いものから表示します。（期限日の昇順）
  - 期限日が過ぎている行は、目立つ色で表示します。
  - 奇数行と偶数行を色分けして、わかりやすいように表示します。
- 作業登録ボタン
  - 作業登録ページを表示します。
- 検索
  - 下記の項目を検索します。
    - 項目名
    - 担当者姓名
    - ~~登録日、期限日、完了日~~ 登録日、期限日、完了日は不要です。
  - 検索ボタンをクリックすると、上記の検索条件に該当するデータを「検索結果ページ」に表示します。



# UI設計（画面設計）作業一覧

- 操作ボタン
  - 完了
    - 該当する行の作業を完了にします。
- 更新
  - 該当する行の「更新ページ」を表示します。
- 削除
  - 該当する行の「削除確認ページ」を表示します。

# UI設計（画面設計）作業登録

作業登録

localhost/todo/entry.html

☆

W

01010011

作業登録

ようこそ〇〇さん

ログアウト

ここにエラーの内容を表示します。

項目名	<input type="text" value="登録内容をここに入力します"/>
担当者	<input type="text" value="テスト1"/>
期限	<input type="text" value="2019/05/10"/>
完了	<input checked="" type="checkbox"/> 完了

登録

キャンセル

# UI設計（画面設計） 作業登録

- 作業登録
  - 新規作業を登録します。
- 入力値のバリデーション
  - 適切なバリデーションを行います。
    - 項目名
      - 100文字以下であること。
    - 担当者
      - 担当する担当者であること。
    - 期限
      - 正しい日付であること。
    - 完了
      - 完了のとき、1であること。
- バリデーションエラー
  - バリデーションでエラーのときは、「作業登録ページ」に戻り、エラーの内容を表示します。
- 例外発生時
  - データベース登録時に例外が発生したときは、「エラーページ」を表示します。

# UI設計（画面設計）作業更新

作業更新

localhost/todo/edit.html

☆

W

01010011

作業更新

ようこそ〇〇さん ログアウト

ここにエラーの内容を表示します。

項目名

登録内容をここに入力します

担当者

テスト1

期限

2019/05/10

完了

☒ 完了

更新

キャンセル

# UI設計（画面設計） 作業更新

- 該当データの検索
  - 該当する作業項目をデータベースから検索し、登録内容を更新します。
- 入力値のバリデーション
  - 適切なバリデーション（値の妥当性チェック）を行います。
    - 項目名
      - 100文字以下であること。
    - 担当者
      - 担当する担当者であること。
    - 期限
      - 正しい日付であること。
    - 完了
      - 完了のとき、1であること。
- バリデーションエラー
  - バリデーションでエラーのときは、「作業登録ページ」に戻り、エラーの内容を表示します。
- 例外発生時
  - データベース更新時に例外が発生したときは、「エラーページ」を表示します。

# UI設計（画面設計） 削除確認

削除確認

localhost/todo/delete.html

☆ W V 0101 0011

削除確認

ようこそ〇〇さん ログアウト

ここにエラーの内容を表示します。

下記の項目を削除します。よろしいですか？

項目名	登録内容をここに入力します
担当者	テスト1
期限	2019-05-10
完了	完了

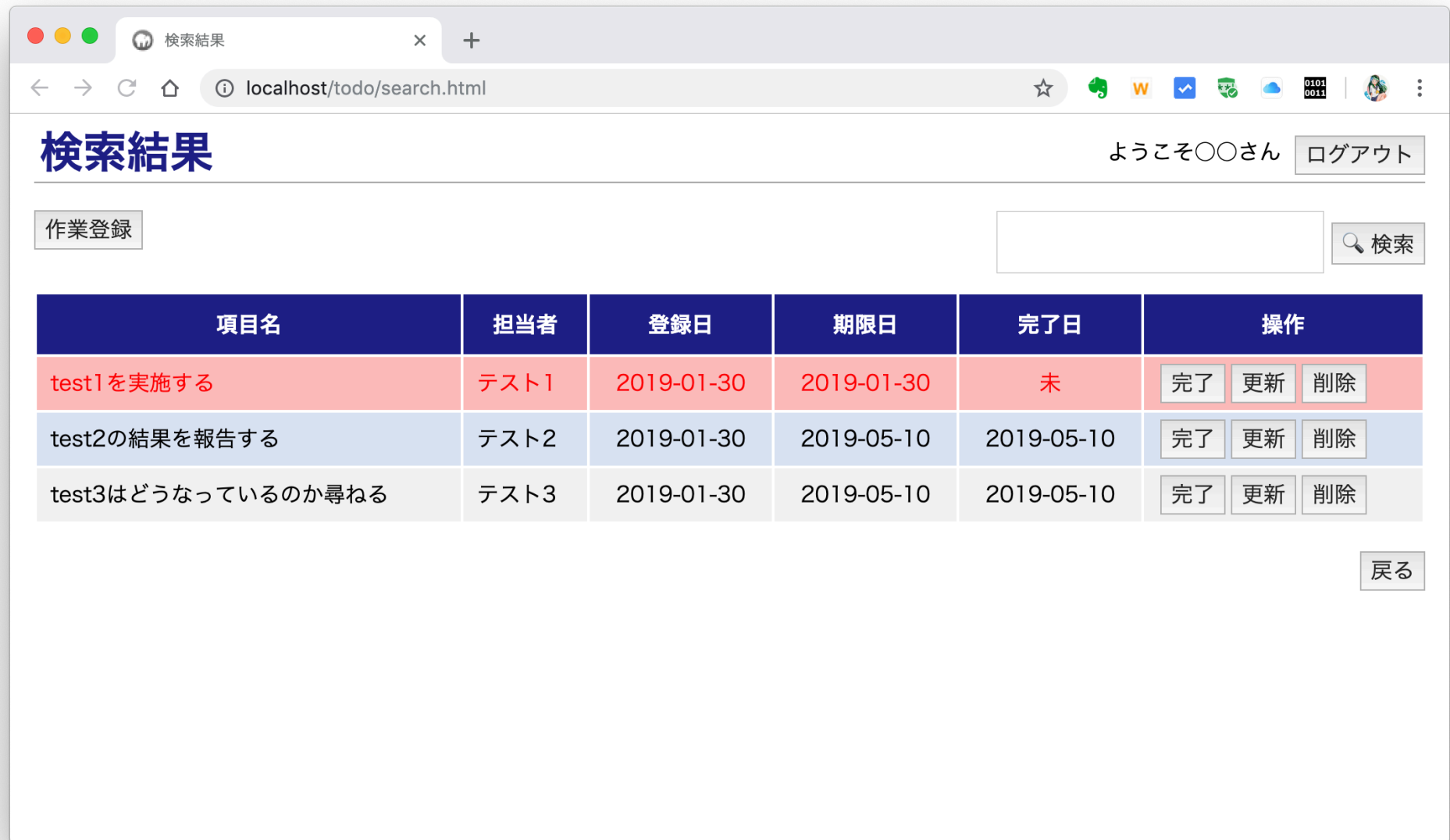
削除

キャンセル

# UI設計（画面設計） 削除確認

- 該当データの検索
  - 該当する作業項目をデータベースから検索し、登録内容を削除します。
- 論理削除する
  - 削除は実際にレコードを削除（物理削除）するのではなく、該当のレコードの「削除フラグ」を「削除済み」にします。
  - 「削除フラグ」が「削除済み」になったレコードは、「作業一覧ページ」、「検索結果ページ」では表示されません。
- 削除ボタン
  - 削除ボタンを押下すると、削除処理が行われ、「作業一覧ページ」が表示されます。
- キャンセルボタン
  - キャンセルボタンを押下すると、直前に表示されていたページに戻ります。
- 例外発生時
  - データベース更新時に例外が発生したときは、「エラーページ」を表示します。

# UI設計（画面設計） 検索結果





# UI設計（画面設計） 検索結果

- 仕様
  - 「作業一覧ページ」に準じます。
- 戻るボタン
  - 戻るボタンを押下すると、「作業一覧ページ」を表示します。

# UI設計（画面設計）エラー



# UI設計（画面設計）エラー

- 例外が発生したとき
  - このページを表示します。
- ログアウトボタン
  - ログアウトボタンを押下すると、ログアウト処理を行い、「ログインページ」を表示します。

# 開発にあたってのポイント(1)

- ログイン
  - ユーザーのレコードの項目をセッション変数に保存し、姓名の表示などに利用します。
  - パスワードは、PHPのpassword\_hash()関数を使ってハッシュ化したものを予め登録します。パスワードのチェックは、ユーザーが入力したパスワードをpassword\_verify()を使ってチェックを行います。  
参照：<https://www.php.net/manual/ja/book.password.php>
- ログアウト
  - ログイン時に保存したセッション変数を削除することで、ログアウト処理とします。
- 新規登録、更新、削除処理
  - 画面表示とデータベースへの登録処理を別のページに分けます。
  - データベースの登録処理の前にバリデーションを行い、エラーがある場合はエラーの内容をセッション変数に保存し、画面表示ページにリダイレクトし、エラー内容を表示します。
  - データベースの登録処理が正常に終了すれば、「一覧表示ページ」にリダイレクトします。
  - 例外が発生したときは、「エラーページ」にリダイレクトします。

# 開発にあたってのポイント(2) クラスを使ってみましょう

開発に余裕があれば・・・

- 機能ごとにクラスを作りましょう。
  - データベース関連
    - データベースに接続するクラス（基底クラス）
    - データベースに接続するクラスを継承したクラス
      - ユーザーのテーブルを操作するクラス
      - 作業項目のテーブルを操作するクラス
- その他のクラス
  - まとめて便利そうなものは、クラスにまとめてみます。また、各ページで使用するものは関数にまとめます。
    - セッションをスタートする関数
    - 入力値をチェックするクラス