

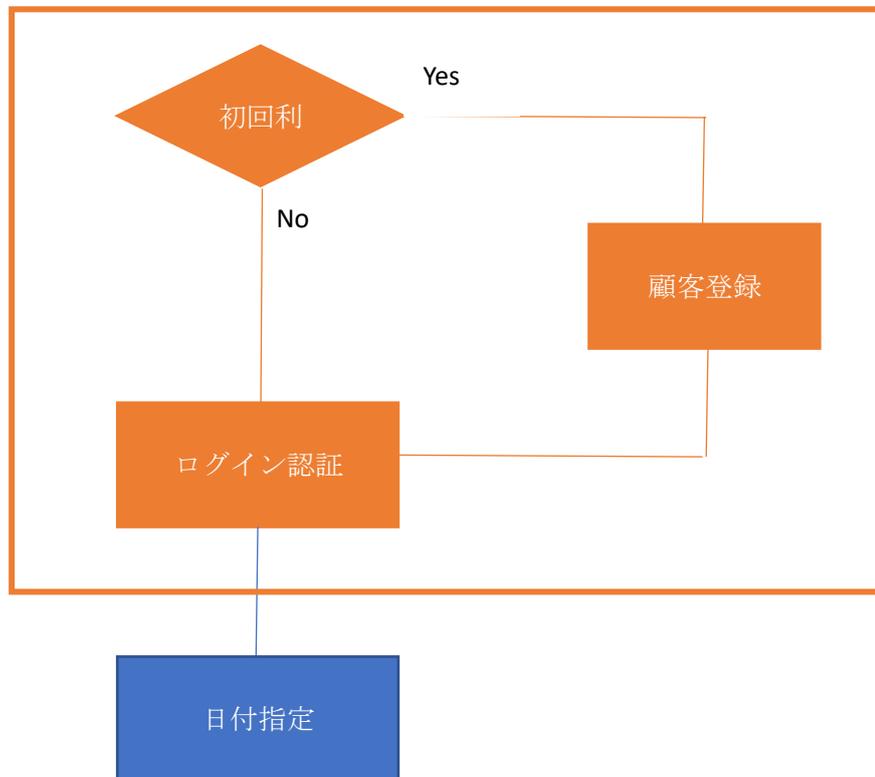
【課題1】

本システムでは、予約登録をする都度、宿泊者情報に「氏名」「連絡先」などを入力する必要がある。多くの Web システムでは、このような個人情報は、顧客情報として別テーブルに登録し、利用者は最初に登録した「ID」「パスワード」を入力することで、何度も入力する手間を省ける仕組みを取っている。

本システムにおいても、同様の機能を実現するための仕組みを検討し対応してください。

● 追加する機能を整理

1. 予約登録の開始で、利用者（宿泊者）が本システムにログインする機能を追加する。
2. 本システムを初めて利用する利用者は、最初に顧客登録する機能を追加する。
3. ログイン後は、ログイン時に取得した個人情報を参照し、宿泊者情報に反映する機能を追加する。

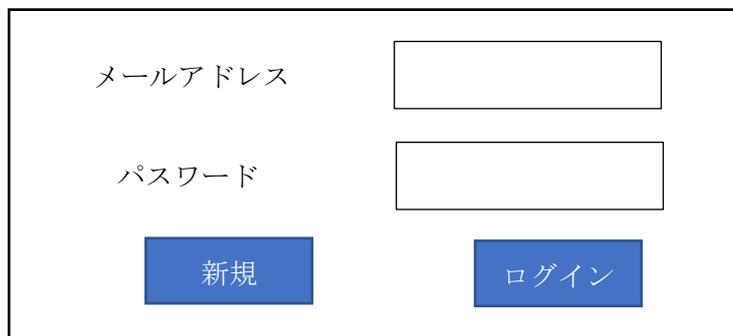


- 修正箇所を整理

1. 顧客のログインページを新規に作成する。(userLogin.php)
2. 顧客のログインチェック機能を新規に作成する。(userLoginCheck.php)
3. 顧客の登録ページを新規に作成する。(customerCreate.php)
4. 顧客の登録チェック機能を新規に作成する。(customerCheck.php)
5. 顧客の登録確認ページを新規に作成する(customerConfirm.php)
6. 顧客をデータベースに追加する機能を新規に作成する(customerInsert.php)
7. その他

- 追加・修正内容を整理

1. 顧客のログインページを新規に作成する。(userLogin.php)



The image shows a login form with two input fields and two buttons. The first input field is labeled 'メールアドレス' (Email Address) and the second is labeled 'パスワード' (Password). Below the first input field is a blue button labeled '新規' (New), and below the second input field is a blue button labeled 'ログイン' (Login).

- ① データベースにログイン ID は別途設けず[メールアドレス]とする。
 - ② パスワードフィールドは「input type="password"」に設定する。
 - ③ [ログイン]ボタンをクリックした場合は「userLoginCheck.php」を実行する。
 - ④ [新規]ボタンをクリックした場合は「customerCreate.php」に遷移する。
2. 顧客のログインチェック機能を新規に作成する。(userLoginCheck.php)
 - ① 入力された「メールアドレス」をキーに[customer]テーブルを読み、登録されている [pass]と一致するかを確認する。
 - ② ①で[pass]が一致した場合は、「氏名」「連絡先」などの顧客情報をセッションに保存し、「reserveDay.php」へ遷移する。
 - ③ ①で[pass]が一致しなかった場合は、「userLogin.php」に遷移する。

3. 顧客の登録ページを新規に作成する。(customerCreate.php)

氏名	<input type="text"/>
連絡先電話	<input type="text"/>
メールアドレス	<input type="text"/>
パスワード	<input type="password"/>
パスワード(確認)	<input type="password"/>
<input type="button" value="戻る"/>	<input type="button" value="確認"/>

- ① [戻る]ボタンをクリックした場合は「userLogin.php」に遷移する。
- ② [確認]ボタンをクリックした場合は、入力された項目をセッションに保存し「customerCheck.php」を実行する。

4. 顧客の登録チェック機能を新規に作成する。(customerCheck.php)

- ① 入力された各項目について以下のチェックをする。
氏名・・・・・・・・・・入力必須
連絡先・・・・・・・・・・入力必須
メールアドレス・・・・入力必須
パスワード・・・・・・・・入力必須
パスワード(確認)・パスワードが一致しているか
- ② 入力項目に不備があればエラーメッセージをセッションに登録し、[customerCreate.php]に遷移する。
- ③ 不備がなければ、[customerConfirm.php]に遷移する。

5. 顧客の登録確認ページを新規に作成する(customerConfirm.php)

氏名	: ○○ △△
電話番号	: 123-456-7890
メール	: xxx@sample.com
パス	: *****
<input type="button" value="戻る"/>	<input type="button" value="確認"/>

- ① セッションに保存されている各項目をページに表示する。
- ② [戻る]ボタンをクリックした場合は「customerCreate.php」に遷移する。
- ③ [確認]ボタンをクリックした場合は、「customerInsert.php」を実行する。

6. 顧客をデータベースに追加する機能を新規に作成する(customerInsert.php)
 - ① セッションに保存されている各項目をデータベースに登録する。
 - ② 登録が完了したら、[reserveDay.php]に遷移する。

7. その他
 - ① [reserveDetail.php]に、セッションに保存された個人情報を初期値としてセットされるように修正を加える。
 - ② [トップページ]からのリンク([ご予約]をクリックした際の遷移先)を変更する。
 - ③ データベースの修正[customer]テーブルに[password]フィールドを追加する。

【課題 2】

管理者ページで表示されている「予約一覧」を CSV 出力する機能を追加してください。
※CSV 出力する手段については、各自ネット検索などで調べて下さい。

● CSV 出力する手段の検討

課題解決には、なにはともあれ「インターネット」の活用です。そこで、まずはインターネット検索で「PHP csv」と検索しましょう！

☆ csv ファイルを作る方法

fopen()関数でファイルポインタを取得する必要がある

fputcsv()関数で csv ファイルを作る事が可能。

☆ ファイルをダウンロードする方法

header()関数で HTTP ヘッダを設定する必要がある。

readfile()関数でファイルが取得可能

● 追加する機能を整理

1. 管理者の予約一覧ページに[csv]ボタンを追加する。
2. [csv]ボタンをクリックした際に csv を作成しダウンロードする機能を作成する。

● 修正箇所を整理

1. 予約一覧ページ (ownerReserveList.php) に [csv]ボタンを追加する。
2. CSV ファイルを出力する機能を新規に作成する。(ownerCSV.php)

● 追加・修正内容を整理

1. 予約一覧ページ (ownerReserveList.php) に [csv]ボタンを追加する。

① 予約一覧の見出しの右端に[csv]ボタンを追加する。

宿泊日付	...	顧客名	メッセージ	CSV
------	-----	-----	-------	-----

② [csv]ボタンをクリックした場合は、「ownerCSV.php」を実行する。

2. CSV ファイルを出力する機能を新規に作成する。(ownerCSV.php)

- ① Csv に出力する予約情報をデータベースから読み込んで配列変数にセットする
- ② fopen()関数を実行しファイルポインタを取得する
- ③ 取得した配列変数とファイルポインタを引数に、fputcsv()関数を実行し csv ファイルを作成する。
- ④ header()関数を実行し http ヘッダを設定する
- ⑤ freadfile()関数を実行しファイルを出力する