

データベース及び演習 期末レポート

K20089 西宮 銀河^{†1,a)}

概要：本稿は、データベースおよび演習の授業における最終課題についてのレポートである。今回はかんたん備忘録という共有可能な軽い TODO リストを作成した。機能の詳細は以下より述べる。

K20089 NISHIMIYA GINGA^{†1,a)}

1. 機能概要

このプログラムの大まかに仮登録からのメール送信, そして本登録完了の流れと管理者側では会員一覧の CRUD, 検索がメインとなっている。

本登録用のデータベースにログイン情報が記載されていない場合, 新規登録をする必要がある。なのでその際には新規登録画面で仮登録をし, 届いたメールに記載されている URL に飛ぶことで本登録が完了となる。また, 新規登録が完了した後はログインページにてログインすることで会員登録情報の修正と退会ができるようになる。修正ページでは仮登録で入力した情報を更新することができ, 退会ページではデータベースから情報を消すことで退会処理をすることができる。またお知らせ機能も備わっており, ログイン後の会員トップページの上部にお知らせを表示することもできる。

管理者ページでは本登録されたユーザらの情報の確認することができ, さらにデータの検索, 新規登録, 更新, 削除が可能となっている。また, それに加えてお知らせの更新を行う機能も備わっている。しかし検索する際は氏名からの検索のみとなっている。また, データの数が多すぎる場合はページャーを使って自動的に番号のページで分けられるようになっている。タイムアウトも実装されているので, 20 分間操作が行われなかった場合は自動的にセッションを削除してログアウトする機能が備わっている。

2. 利用技術

2.1 PHP

PHP とは”PHP:Hypertext Preprocessor”の略で, 直接 HTML に埋め込んで使用することができる汎用プログラミング言語である。構文は C 言語, Java, Perl から派生しており, それに加えて PHP 独自の構文を組み込むことで PHP 特有の機能を使用できる。PHP はサーバサイド言語であるので, 主に Web サーバ側で動作する。なのでクライアント側からブラウザが HTTP リクエストを送ったとすると, サーバ側が PHP を処理して生成された HTML 文書などをクライアント側に返すという仕組みとなっている。そして PHP はクロスプラットフォーム言語なので Linux, Windows, MacOS はもちろん, NginX や Apache など全ての web サーバ, さらに Microsoft Azure や Amazon AWS などのクラウド環境にも対応している。

また現在の最新バージョンは PHP 8 であり, これは Zend Engine 4 という PHP の実行エンジンの最新バージョンを使用している。これによってオブジェクト指向プログラミングの機能が使えるようになっている。

2.2 MariaDB/MySQL

まず, MySQL は 1995 年に開発されたオープンソースソフトウェアのリレーショナルデータベースである。'95 年以降は所有者も管理者も転々と変わっていたが 2010 年に所有権が Oracle 社に移るが, 所有権が渡った今でもオープンソースであることは変わっていない。

MySQL はリレーショナルデータベースと呼ばれるデータの保存方法が使われている。リレーショナルデータベースとは 1 つのテーブルに全ての情報を挿入するのではなく,

^{†1} 現在, 愛知工業大学
Presently with AICHI INSTITUTE OF TECHNOLOGY
University
^{a)} gingin.sol@pluslab.org

複数のテーブルに分けてそれぞれを関連づけて管理する方法である。例えば名前や住所などの顧客データと購入製品や価格などの注文データを扱うテーブルを作るとすると、もし一つのテーブルに全てのデータを入れておくとなった際にデータの重複やデータ削除時の混乱が起り、検索が難しくなる。その問題に対処するためにリレーショナルデータベースではキーと呼ばれるユニークなIDを利用して異なるテーブル同士を繋げ、データの取り出しや削除などが容易になっている。

そして、MariaDBはこのMySQLをベースに作られたデータベースである。MySQLと比べてパフォーマンス、安定性、開示性が高く、もっとも有名なリレーショナルデータベースの一つとされている。MySQLとの完全な互換性、GPLの採用、さらには開発者コミュニティが豊富なのでMariaDBはMySQLの上位互換と言える。

2.3 CSS

CSSとは”Cascading Style Sheets”の略で、HTMLやXML文書の見栄えなどを整えるために使われるスタイルシート言語である。HTML内で指定されたタグやidなどをセレクトで指定して色や段落などの設定ができる。

2.4 Smarty

SmartyとはPHP用のテンプレートエンジンである。図1に示すようにプロジェクトファイルをデータベースと接続するモデル、HTML表示を行うためにクライアントにレスポンスを返すビュー、これらモデルとビューをつなぐコントローラの3つの役割に分けることでフロントエンドとバックエンドそれぞれの開発がしやすくなっている。デリミタの使用ができたり、構文はPHPパーサが処理するので複雑な条件式も入力でき、さらに高速で動作できることが特徴となっている。

2.5 HTTP

HTTPとは”Hypertext Transfer Protocol”の略でハイパーテキストを転送するためのアプリケーション層プロトコルである。主にサーバ側とクライアント側との通信の目的のために使われるが別の用途でも使用されることがある。基本的な動作としてはクライアント側がサーバ側にリクエストを送信するためにポートを開き、サーバ側からレスポンスが帰ってくるまで待機するという仕組みとなっている。また、HTTPは主にTCP/IP層の上の通信で使われるが、UDPのようなトランスポート層でも使用されることがある。

2.6 SMTP/IMAP/POP

SMTPは”Simple Mail Transfer Protocol”の略で、メール送信の際に用いられるプロトコルである。このプロトコ

ルでスパムメールを防ぐためにはSMTP認証が必要である。そのためにはメールアドレスとパスワードの入力でメールサーバに接続することができ、安全なメール送信が可能となる。

IMAPは”Internet Message Access Protocol”の略で、メール受信の際に用いられるプロトコルである。クライアント側からリモートウェブサーバにアクセスしてメールを確認することができるので、複数の端末から同時に接続できたり、端末が変わっても接続することができる。

POPは”Post Office Protocol”の略で、こちらもIMAP同様、メール送信の際に用いられるプロトコルである。IMAPとの違いはリモートサーバ側からクライアントに接続してメールをローカルにダウンロードするのでオフラインでもメールの確認ができるようになっている。さらにダウンロードした後はサーバに残っているメッセージは削除されるのでサーバの容量の削減になることが特徴となっている。

3. システム設計

3.1 システム概要

Smartyを使って、コントローラでデータベースと表示を操作している。実際会員ページではMemberController.php、管理者ページではSystemController.phpを使ってモデルから受け取ったデータベースの情報をビューに送り、表示させている。中で使用されているコントローラは全てBaseController.phpというベースとなるコントローラを継承しており、BaseControllerのログイン状態やエラー表示の機能に加えて、それぞれのコントローラで使用される機能を追加している。

モデルも同様にBaseModel.phpというベースファイルを継承してそれぞれの場面で使用されるモデルクラスを生成している。このクラスではデータベースに接続して様々な操作をする機能を持っており、このプロジェクトでは新規登録時のデータベースへの挿入や退会時のデータベースからの削除、管理者画面でのデータの検索などをする際にモデルの役目が果たされていると言える。

また、ビューで表示するHTML文書は拡張子が.htmlではなく、.tplというテンプレートファイルを使って表示している。これを使えばif構文などが使用できるのでコントローラから受け取った値などをhtmlに便利に表示することができる。

3.2 画面遷移

まず、会員ページでの状態遷移図は図2のような構成となっている。ログイン画面からは新規登録画面とログインした場合の会員のトップ画面に飛ぶことができる。新規登録画面ではメールアドレスなど必要な情報を入力してもらい、その次に登録情報の確認に移る。そしてメールの送信が完了し、登録したメールアドレスに送られた専用メー

ルに書かれたリンクに飛ぶと、登録完了となり、ログイン画面に戻ることができる。この際にすでに有効期限などが切れ、無効となったリンクに飛ぼうとするとエラーとなり、同じくログイン画面に戻ることになる。

会員登録が済んでいる場合はログイン画面で会員情報を入力して会員トップ画面に飛ぶことができる。会員トップ画面では登録情報の変更と退会をすることができ、またトップページからログアウトをすることも可能である。更新画面では新規登録時と同じ入力フォームが表示され、データの変更ができる。次に確認画面で確認を行い更新完了となる。そして退会画面では本当に退会してもいいかの削除確認画面が表示され、よければ削除完了となりログアウトされ、ログイン画面に戻る。

次は管理者ページでの状態遷移図は図3のようになっている。まず、管理者用のユーザ名とパスワードを入力すると管理トップ画面に行くことができる。管理トップ画面からは会員一覧の表示に飛ぶことができ、そこでは会員の新規登録、更新、削除を行うことができる。

新規登録を行う際はまず登録情報の入力、そして確認画面にて確認をすることで登録を完了することができる。更新画面でも同様に登録情報の変更を行い、確認をして更新が完了される。削除画面では本当に削除していいかの表示の確認の後、会員の削除を行うことができる。

3.3 データベース設計

本システムで使用されているデータベースでは次の5つのテーブルが使われている。

- 会員情報テーブル
- 県名テーブル
- 仮登録テーブル
- 管理者情報テーブル
- お知らせテーブル

会員情報テーブルでは本登録された会員情報を保持し、県名テーブルでは会員情報入力に必要な都道府県名を保持している。仮登録テーブルでは仮登録の際にメールを送信したアカウント情報が保持されている。このテーブルでは本登録が完了すれば該当データは自動的に削除されるようになっている。そして管理者情報テーブルでは会員ではない管理者の情報を保持するのに使用されている。管理者ページにログインする際はこちらのテーブルから参照される。最後にお知らせテーブルはお知らせの文章を保持するために使われている。

会員情報テーブルのテーブル構成は表1のようになっている。一意の数値を挿入するためのidフィールドがまず最初にあることが分かる。このフィールドは3バイトの整数型を取扱い、数値が無であることは許されており、さらに自動繰上げ機能が付いている。これによってデータが挿入されると自動的にidの値を繰り上げることが可能と

なっている。次に続くusernameフィールドからfirst_nameフィールドに関してはユーザネーム、パスワード、名字、名前を示しており、括弧内の文字数分を確保した文字列を格納している。birthdayフィールドは誕生日を示し、8桁の文字型で格納されている。kenフィールドでは2バイトの整数型を取り扱い、県名テーブルと連携して都道府県のそれぞれのidが入ることになっている。reg_dateフィールドでは日付型を扱い、登録日時を格納している。

表 1 会員情報テーブル構成

Field	Type	Null	Key	Default	Extra
id	mediumint unsigned	NO	PRIMARY	NULL	auto_increment
username	varchar(50)	YES		NULL	
password	varchar(128)	YES		NULL	
last_name	varchar(50)	YES		NULL	
first_name	varchar(50)	YES		NULL	
birthday	char(8)	YES		NULL	
ken	smallint	YES		NULL	
reg_date	datetime	YES		NULL	
cancel	datetime	YES		NULL	

次に県名テーブルである。県名テーブルの構成は表2の通りである。同じく一意の数値が格納されるidフィールドがあるが、今回は47都道府県分の番号を入れるのみなのでsmallint型を使用して2バイトの数値を扱うようにしている。また、kenフィールドでは10文字の都道府県名が格納されるようになっている。

表 2 県名テーブル構成

Field	Type	Null	Key	Default	Extra
id	smallint	NO	PRIMARY	NULL	
ken	varchar(10)	YES			

次は仮登録テーブルである。仮登録テーブルの構成は表3に示す通りである。このテーブルの構成はほとんど会員情報テーブルとほぼ同じだが、仮登録完了メールに書かれたリンクに飛ぶ際に使用されるパスワードを格納するlink_passフィールドが追加されている。このパスワードを使用して本登録が完了したかなどの判別を行なっている。

表 3 仮登録テーブル構成

Field	Type	Null	Key	Default	Extra
id	mediumint unsigned	NO	PRIMARY	NULL	auto_increment
username	varchar(50)	YES		NULL	
password	varchar(128)	YES		NULL	
last_name	varchar(50)	YES		NULL	
first_name	varchar(50)	YES		NULL	
birthday	char(8)	YES		NULL	
ken	smallint	YES		NULL	
link_pass	varchar(128)	YES		NULL	
reg_date	datetime	YES		NULL	

最後にお知らせテーブルである。お知らせテーブルの構成は表4に示すものである。今までと同じように一意なidを最初にフィールドとして定義している。次にお知らせの件名であるsubjectフィールドを256文字までという制限とともに定義している。また本文であるbodyフィールドにはtext型を宣言しているので自由な文字列が入力と可能

となっており、最後にはお知らせがされた時間を挿入する reg_date が定義されている。

表 4 お知らせテーブル構成

Field	Type	Null	Key	Default	Extra
id	mediumint unsigned	NO	PRIMARY	NULL	auto.increment
subject	varchar(256)	YES		NULL	
body	text	YES		NULL	
reg_date	datetime	YES		NULL	

3.4 システム詳細

会員ページのメインファイルとなる index.php ではまず、初期化ファイルの init.php を呼び出し、データベースのユーザ設定やライブラリなどの初期設定を行なっている。次に MemberController.php という会員情報を取り扱うコントローラクラスを実体化して、更に Auth クラスの機能を使用してログインされている状態かを判別することで表示するページを変えている。MemberController で使われるページ遷移の状態には次のような 6 つの種類がある。ちなみにこれらのステートメントは URL の末尾につけられるので状態の判別ができるようになっている。

- ログイン時
 - － デフォルト状態：会員トップページ
 - － "logout"：ログアウトされたとき
 - － "modify"：登録情報が変更されたとき
 - － "delete"：退会するとき
- 未ログイン時
 - － デフォルト状態：ログインページ
 - － "regist"：新規登録するとき
 - － "authenticate"：登録データが条件に合っているか判別するとき

またこのコントローラは BaseController というクラスを継承しており、このクラスでは Smarty の実装や会員情報入力時のルールなどについて定義している。このプロジェクトで使用されるコントローラは全てこのベースコントローラを継承している。

そして NoticeController ではお知らせに関するデータを扱っている。まず NoticeModel クラスを使用してお知らせデータベースからデータを持ってくることでお知らせの文章を取り出すことができる。あとはテンプレートファイルで整った表示をすることでお知らせをデータベース上のお知らせ情報を表示させることが可能となっている。

次の PrememberController は仮登録メールから本登録を行う際に用いられている。PrememberModel という仮登録データベースを扱うモデルクラスを使用して、リンクに記載されているリンクパスワードの一致とユーザの名前が一致していれば本登録完了の文を表示し、一致しなかったりリンクにパラメータがなければエラーを表示することで本登録処理を行なっている。

最後に SystemController である。これは管理者画面で使われるコントローラである。Auth クラスを使用してセッション開始と認証を行い、まだ認証がされていない場合は状態を変えてログインページに飛ぶようになっている。また、MemberController 同様、ページ遷移のためのパラメータを扱っており、次の 9 つのステートメントがある。

- デフォルト状態：ログインページまたは管理者ページ
- "login"：ログインページにいるとき
- "logout"：ログアウトされたとき
- "modify"：会員情報を変更するとき
- "delete"：会員情報を削除するとき
- "list"：会員情報を表示しているとき
- "regist"：会員情報を新規で登録するとき
- "notice"：お知らせ表示を編集するとき
- "authenticate"：登録データが条件に合っているか判別するとき

これらの状態に応じて表示するテンプレートファイルを切り替えている。

3.5 変数設計

本システムで使われる変数の多くは主に BaseController で宣言されている。まず type 変数と action 変数はそれぞれ状態遷移のときに使われている。type 変数はそれぞれのページにおける "login" や "modify" などの状態の名前を保持する役割を持っている。action 変数は何かしらの動作を行おうとしている時にその動作の名前を保持するものである。この場合の例ではフォーム入力するときの "form"、確認画面であることを示す "confirm"、動作が完了したことを示す "complete" などがある。次の next_type 変数と next_action 変数は名前の通り次に遷移する状態を保持する。

file 変数では使うテンプレートファイル名を格納している。ここにファイルを定義することで HTML を表示している。

次の form 変数ではフォーム入力する際に用いられる HTML 要素が格納されている。ここに text フォームや submit フォームを追加することでそれに応じたフォーム要素が使用できるようになる。このとき HTMLQuickForm2 という入力フォームを生成するクラスが使用されているので Smarty と組み合わせでテンプレート 1 枚で入力フォームが便利に生成可能となっている。この際に renderer 変数に HTMLQuickForm2Renderer というクラスを割り当てて Smarty との組み合わせが可能となっている。また、auth 変数には Auth クラスを定義して認証機能が使用できるようになっており、さらに view 変数には Smarty クラスを定義することで Smarty の機能が使用できる。

次の title 変数では各ページの見出し表示、message 変数

では内容文を格納し, `auth_error_mess` 変数には認証失敗時の表示をし, `debug_str` 変数ではデバッグ用の表示をしている。最後の `login_state` 変数では会員でログインされているか, 管理者でログインされているかの状態を保持している。

4. 実装

4.1 実装環境

本システムの実装環境は以下の通りである。

- XAMPP 7.4.28
- MariaDB 10.4.21
- PHP 8.1.4

4.2 環境設定

使用している XAMPP はデフォルト設定なので使用しているポート番号は 80 であり, サーバネームは "localhost" という設定となっている。また, データベース設定としてはデータベース接続用のユーザネームは "root", パスワードは未設定, ホスト名は "localhost", データベース名は "memberdb" となっている。

4.3 動作検証

新規登録までの画面遷移は図 4 のようになっている。3.2 でも述べたようにログイン画面から新規登録画面に飛ぶことで必要情報を入力して, 入力情報を確認した後に仮登録完了画面に飛ぶようになっている。

次に図 5 は会員登録後, ログインを完了させるまでの画面遷移である。先ほど登録した情報をフォームに入力するとログインが完了となり, 登録した名前が表示されるようになる。

最後に管理者画面でのログインの画面遷移を図 6 に示している。まず管理者用のユーザネームとパスワードを入力すると管理者トップページに飛ぶことができる。さらにそこから会員一覧画面に飛ぶことができ, そこでは登録情報の新規登録, 変更, 削除, 検索ができるようになっている。

5. まとめ

今回は教科書 9 章・10 章のサンプルプログラムについてまとめた。普段フレームワークを使用する際はテンプレートを編集して開発することが多かったのですが, Smarty を使用した MVC 方式の設計を全て自分で書くことはあまりなく, どういった形式でこのプログラムが書かれているかをしっかり理解することができた。他の MVC 方式のフレームワークを使用する際にも今回学んだ知識を活かして開発を進めていきたいと感じた。

参考文献

- [1] PHP Group . "一般的な情報" . php . <https://www.php.net/manual/ja/faq.general.php> .

- (閲覧日:2022/05/29)
- [2] PHP Group . "Zend API: PHP のコアをハックする" . php . <http://php.adamharvey.name/manual/ja/internals2.zel.zendapi.php> . (閲覧日:2022/05/29)
- [3] PHP TUTORIAL . "What is PHP" . phptutorial.net . <https://www.phptutorial.net/php-tutorial/what-is-php/> . (閲覧日:2022/05/29)
- [4] KINSTA . "MySQL とは? 初心者にわかりやすい説明" . KINSTA . (更新日:2020/07/03) . <https://kinsta.com/jp/knowledgebase/what-is-mysql/> . (閲覧日:2022/06/01)
- [5] MariaDB Foundation . "MariaDB Server: The open source relational database" . MariaDB Foundation . <https://mariadb.org/> . (閲覧日:2022/06/01)
- [6] Mark Smallcombe . "MariaDB vs MySQL: 徹底比較" . integrate.io . (更新日:2020/09/03) . <https://www.integrate.io/jp/blog/mariadb-vs-mysql-everything-you-need-to-know-ja/> . (閲覧日:2022/06/01)
- [7] MDN contributors . "CSS: カスケーディングスタイルシート" . mdn web docs . (更新日:2021/07/18) . <https://developer.mozilla.org/ja/docs/Web/CSS> . (閲覧日:2022/06/01)
- [8] smarty . "Smarty とは?" . smarty . <https://www.smarty.net/docsv2/ja/what.is.smarty.tpl> . (閲覧日:2022/06/01)
- [9] MDN contributors . "HTTP" . mdn web docs . (更新日:2021/09/18) . <https://developer.mozilla.org/ja/docs/Web/HTTP> . (閲覧日:2022/06/01)
- [10] SiteGround Hosting Ltd . "Email Protocols - POP3, SMTP and IMAP Tutorial" . SiteGround . <https://www.siteground.com/tutorials/email/protocols-pop3-smtp-imap/> . (閲覧日:2022/06/01)
- [11] SiteGround Hosting Ltd . "What is SMTP?" . SiteGround . <https://www.siteground.com/kb/what-is-smtp/> . (閲覧日:2022/06/01)