

調整くん

1. ソフトウェア構成

本システムは、打ち合わせや宴会など多数の人が参加するイベントの日程を決める際に、参加者の都合を Web 上で容易に確認できるようにすることで、イベントの日程調整の作業を支援するシステムである。本仕様書はサーバ側の仕様について説明する。

図1にサーバ側のソフトウェア構成図を示す。

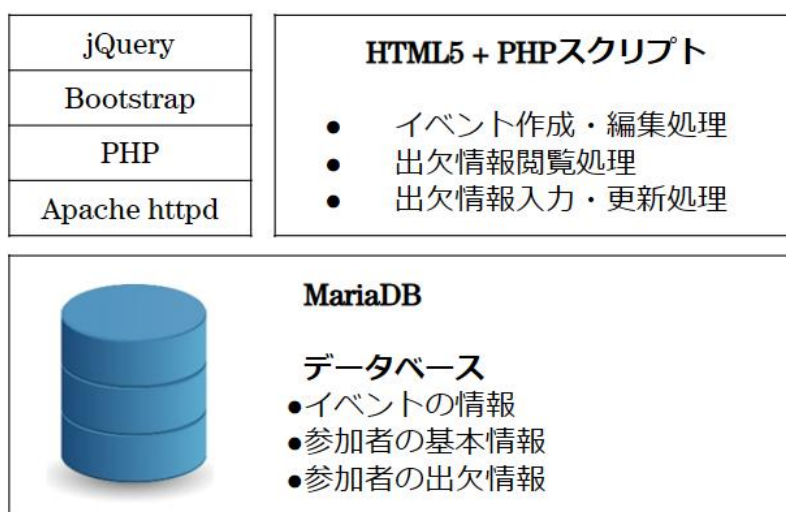


図1 ソフトウェア構成図

使用するソフトウェアのバージョンは次の通りである。

- MariaDB 10.4.32
- Apache httpd 2.4.58
- PHP 8.2.12
- Bootstrap 5.3.2
- jQuery 1.12.4

2. モジュール構成

ソフトウェアのモジュール構成を図2に示す。

画面	制御	モデル
① index.php イベント作成画面	modify_event.php イベント作成、編集	access_db.php イベント作成、編集
④ view_modify_event.php イベント編集画面	check_event.php イベント情報の取得 delete_event.php イベント削除	イベントIDから イベント情報を得る イベントIDから イベントを削除 参加者情報追加、更新 イベントIDから 参加者情報を得る
② view_url.php URL(ID)通知画面		
③⑤ view_event.php 出欠情報閲覧画面	view_attendee.php 参加者情報の取得	
⑥⑦ view_modify_attendee.php 出欠作成画面	create_attendee.php 参加者情報入力 のフォームを設置 modify_attendee.php 参加者情報更新 のフォームを設置 post_form.php 参加者情報を登録 post_form_modify.php 参加者情報を更新	

図2 モジュール構成

ここに①～⑦の画面は、外部仕様書図3に示す画面遷移図の画面に対応している。

また、図3に画面描画（ビュー）と制御（コントローラ）のモジュールについて、遷移の関係を示す。

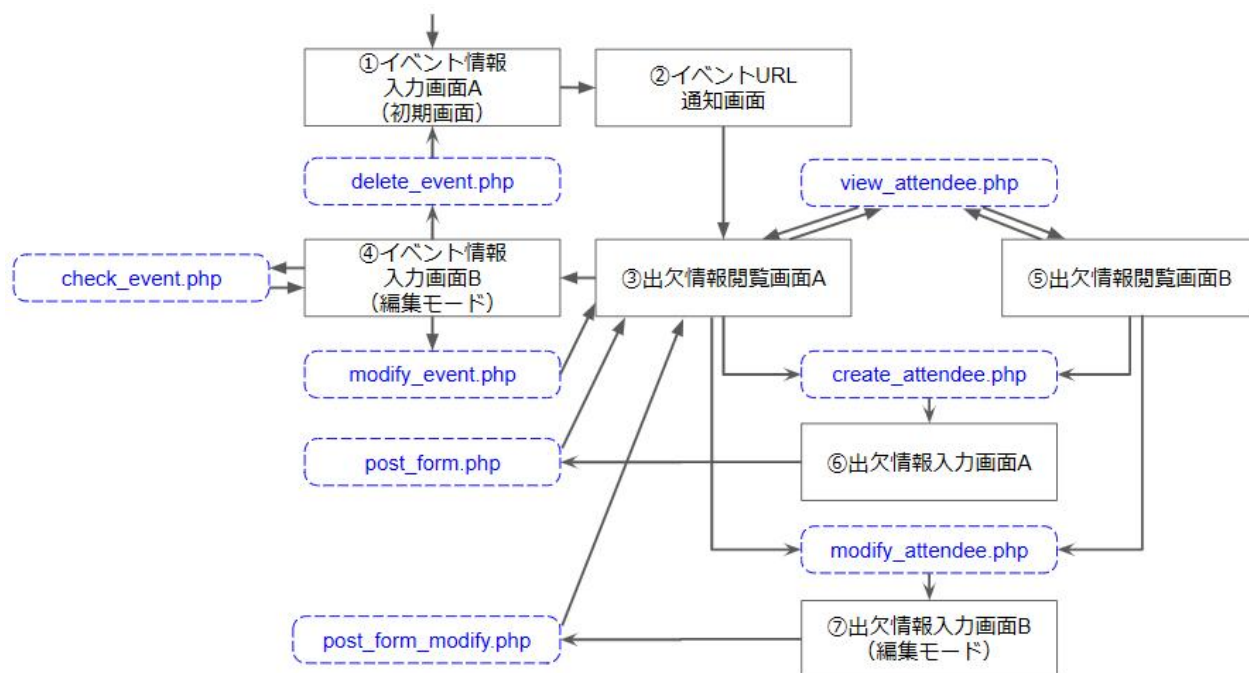


図3 画面遷移とモジュールの関係

3. データベース仕様

MySQL 上のデータベースは、次の 3 つのテーブルを持つ。

- event_info 登録したイベントの情報を保持する
- attendee_info 参加者に関する情報を保持する
- attendee_status 出欠状態に関する情報を保持する

表 1 ～ 3 に各々のテーブルのカラムの情報を示す。

表 1 テーブル event_info のカラム

データ名	意味	型
event_id	自動的に生成したイベントの ID	char(20), not null, primary key
event_name	イベントの名前	char(100)
dates	イベントの時間	char(300), not null
memo	イベントのメモ	char(300)

表 2 テーブル attendee_info のカラム

データ名	意味	型
event_id	イベントの ID	char(20), not null, primary key
attendee_name	参加者の名前	char(100), primary key
comment	参加者のコメント	char(300)

表 3 テーブル attendee_status のカラム

データ名	意味	型
event_id	イベントの ID	char(20), not null
attendee_name	参加者の名前	char(100)
date	参加時間	char(100)
status	参加できるかどうかを表す数字 [×->0、△->1、○->2]	int, not null

4. モジュール仕様

4.1 モジュール `access_db.php`

このモジュールは MVC モデルの M（モデル）部分を担当し、データベースへのアクセスと直接操作を行う関数を含んでいる。他のモジュールから `include` 文で本ファイル全体を読み込む事によって、下に説明する関数群が使用可能になる。

4.1.1 関数 `init_db`

引数	無し
戻り値	無し
機能	
MySQL サーバーに接続し、データベースを使用可能にする。他の関数の呼び出しに先立って実行されなければならない。	

4.1.2 関数 `get_event_info_from_event_id`

引数	イベント ID（整数）
戻り値	エラーコード（整数） 0: 正常終了。イベント ID を取得できた 1: 異常終了。イベント ID を取得できなかった
機能	
引数で与えられたイベント ID をテーブル <code>event_info</code> に対応するイベント情報に検索し、正しく取得出来た場合には、以下のグローバル変数に格納する。 <code>global_event_name</code> <code>global_event_dates[]</code> 得られた文字列を改行記号で分割した配列にする <code>global_event_memo</code>	

4.1.3 関数 `create_event`

引数	イベント名（文字列）、日程候補（文字列）、メモ（文字列）
戻り値	エラーコード（整数） 0: 正常終了。イベントを登録した 1: 異常終了。イベントを登録しなかった
機能	
引数で与えられたイベント名、日程候補（改行記号を含む）、メモを持ちながら、新しいイベント ID を生成し、情報をテーブル <code>event_info</code> に登録する。登録成功の場合には、イベント ID をグローバル変数に格納する。 <code>global_event_id</code>	

4.1.4 関数 `modify_event`

引数	イベント ID（整数）、イベント名（文字列）、日程候補（文字列）、メモ（文字列）
戻り値	エラーコード（整数） 0: 正常終了。イベント ID を更新した 1: 異常終了。イベント ID を更新しなかった
機能	
引数で与えられたイベント ID に対応するイベント情報をテーブル <code>event_info</code> に検索し、イベント名、日程候補、メモを更新する。	

4.1.5 関数 `delete_event_from_event_id`

引数	イベント ID (整数)
戻り値	エラーコード (整数) 0: 正常終了。イベントを削除した 1: 異常終了。イベントを削除しなかった
機能	
引数で与えられたイベント ID をテーブル <code>event_info</code> に対応するイベント情報を削除する。同時に、テーブル <code>attendee_info</code> 、 <code>attendee_status</code> もイベント ID に対応する情報を削除する。	

4.1.6 関数 `get_attendee_info_list_from_event_id`

引数	イベント ID (整数)
戻り値	エラーコード (整数) 0: 正常終了。参加者情報を取得できた 1: 異常終了。参加者情報を取得できなかった
機能	
引数で与えられたイベント ID をテーブル <code>attendee_info</code> に対応する参加者名 (文字列)、コメント (文字列) を検索し、正しく取得出来た場合には、グローバル変数 <code>global_attendee_names[]</code> 、 <code>global_attendee_comments[]</code> に格納する。	

4.1.7 関数 `get_attendee_status_list_from_event_id_and_event_date`

引数	イベント ID (整数)、参加時間 (文字列)
戻り値	エラーコード (整数) 0: 正常終了。ステータスを取得できた 1: 異常終了。ステータスを取得できなかった
機能	
引数で与えられたイベント ID と参加時間をテーブル <code>attendee_status</code> に対応する参加者の <code>status</code> を検索し、正しく取得出来た場合には、グローバル変数に格納する。 <code>global_attendee_statuses[]</code> (整数の配列)	

4.1.8 関数 `create_attendee_info`

引数	イベント ID (整数)、参加者名 (文字列)、コメント (文字列)
戻り値	エラーコード (整数) 0: 正常終了。参加者情報を登録した 1: 異常終了。参加者情報を登録しなかった
機能	
引数で与えられたイベント ID、参加者名とコメントをテーブル <code>attendee_info</code> に追加する。	

4.1.9 関数 create_attendee_status

引数	イベント ID（整数）、参加者名（文字列）、参加時間（文字列）、ステータス（整数）
戻り値	エラーコード（整数） 0: 正常終了。参加情報を登録した 1: 異常終了。参加情報を登録しなかった
機能	
引数で与えられたイベント ID、参加者名、参加時間、とステータスをテーブル <u>attendee_status</u> に追加する。	

4.1.10 関数 modify_attendee_info_from_event_id_and_attendee_name

引数	イベント ID（整数）、元の参加者名（文字列）、変更後の参加者名（文字列）とコメント（文字列）
戻り値	エラーコード（整数） 0: 正常終了。参加者情報を更新した 1: 異常終了。参加者情報を更新しなかった
機能	
引数で与えられたイベント ID と元の参加者名をテーブル <u>attendee_info</u> に対応する参加者情報を変更後の参加者名とコメントに更新する。	

4.1.11 関数 modify_attendee_status_from_event_id_and_attendee_name

引数	イベント ID（整数）、元の参加者名（文字列）、変更後の参加者名（文字列）、参加時間（文字列）、ステータス（整数）
戻り値	エラーコード（整数） 0: 正常終了。参加情報を更新した 1: 異常終了。参加情報を更新しなかった
機能	
引数で与えられたイベント ID と元の参加者名と参加時間をテーブル <u>attendee_status</u> に対応する参加情報を変更後の参加者名、参加時間と <u>status</u> に更新する。	

4. 2 イベント情報系モジュール

4.2.1 モジュール view_modify_event.php

入力	セッション変数で次の値を受け取る \$_SESSION['event_id'] イベント ID (整数)
機能	
セッション変数でイベント ID を受け取り、相当するイベントが存在しない場合は、①イベント作成画面を描画を行う。 相当するイベントが存在する場合は④イベント情報をフィールドに入れ、「出欠表をつくる」ボタンを「更新する」ボタンに変わる。 ボタンがクリックされると、modify_event.php へ遷移する。「つくるか更新か」を modify_mode (整数)、イベント名を event_name (文字列)、日程候補を event_dates (配列)、メモを event_memo (文字列)、POST メソッドで渡す。	

4.2.2 モジュール modify_event.php

入力	セッション変数と POST メソッドで次の値を受け取る \$_SESSION['event_id'] イベント ID (整数) \$_POST['modify_mode'] つくるか更新か (整数) \$_POST['event_name'] イベント名 (文字列) \$_POST['event_dates'] 日程候補 (文字列) \$_POST['event_memo'] メモ (文字列)
機能	
POST メソッドで「つくるか更新か」とイベントの情報を受け取り、次のチェックを行う。 ・イベント名と日程候補が空文字列でないか ・イベント ID に相当するイベント ID が存在するか (“つくる”の場合はエラー。更新では逆) エラーを発見すれば、セッション変数 error_msg にエラーメッセージを格納し、前のページに戻り、エラーメッセージを表示する。 エラーが無ければ、view_event.php へ遷移する。	

4.2.3 モジュール check_event.php

入力	セッション変数で次の値を受け取る \$_SESSION['event_id'] イベント ID (整数) もし以上の変数がなかった場合は次の値を受け取る \$_SESSION['event_name'] イベント名 (文字列) \$_SESSION['event_dates'] 日程候補 (文字列) \$_SESSION['event_memo'] メモ (文字列)
機能	
javascript から POST メソッドでイベント情報を要求し、json 形式でデータを渡す。 セッション変数でイベント ID を受け取った場合は、データベースからイベント情報を受け取る。無ければ、セッション変数でイベントの情報を受け取る (modify_php がエラーの場合は設置される)。	

4. 3 出欠情報系モジュール

4.3.1 モジュール view_event.php

(navbar.php, message.php と event_info_and_attendee_list.php 三つのファイルを include で含まれている)

入力	ゲット変数で次の値を受け取る \$_GET['eid'] イベント ID セッション変数で次の値を受け取る \$_SESSION['event_info'] イベント情報 (リスト) \$_SESSION['attendee_info_list'] 参加者情報 (リスト) \$_SESSION['attendee_status_list'] 出欠状態の情報 (リスト) \$_SESSION['view_attendee_form'] フォームの表示状態 (bool) \$_SESSION['attendee_form_action'] フォームの post 制御ファイル名 (文字列) \$_SESSION['modifying_attendee_name'] 変更した参加者名 (文字列) クッキー変数で次の値を受け取る \$_COOKIE['creator_event_id_list'] cookie にある全部のイベント ID (リスト)
機能	
イベント詳細と参加者出欠情報の表示画面、出欠情報の登録や変更フォームの表示画面。 ゲット変数で現在ページのアドレスからイベント ID を受け取り、ゲットできない場合は、message.php 画面を映り、エラーメッセージを表示する。 セッション変数でイベント情報を受け取り、取らない場合は view_attendee.php に戻る。 セッション変数で参加者情報を受け取り、取らない場合は参加者を 0 にする。取った情報を該当する位置に表示する セッションでフォームの表示状態を受け取り、フォームを表示するかどうかを制御する。 セッション変数でフォームのアクション・ファイル名を受け取る。フォームが新規参加者の登録か、既存参加者の情報変更か、違いアクション・ファイルで処理するので、フォームの正しいアクションを設置する。 セッション変数で変更後の参加者名を受け取り、元の参加者名と区別する。 クッキー変数で現在ユーザーが管理しているイベントのリストを受け取り、閲覧しているイベントの管理者かどうかを判断することで、管理者画面或いは参加者名画面を映る。	

4.3.2 モジュール view_attendee.php

入力	セッション変数で次の値を受け取る \$_SESSION['event_id'] イベント ID (文字列)
機能	
<p>view_event.php 画面のコントローラーです。関数を呼び出し、データベースからデータを受け取り、各セッションに設置して、view_event.php に移る。また、不要なセッションをクリアする。</p> <p>セッション変数でイベント ID を受け取り、データベースからイベント情報と参加者情報を取る、event_info(リスト)、attendee_info_list と attendee_status_list 三つのセッションに保存する。</p>	

4.3.3 モジュール create_attendee.php

入力	なし
機能	
<p>view_event.php 画面のコントローラーです。画面を制御する各セッションを設置して、view_event.php に移る。</p> <p>view_event.php 画面のフォームを表示するかの制御変数とフォームのアクション処理ファイルを設置する。</p>	

4.3.4 モジュール modify_attendee.php

入力	ゲット変数で次の値を受け取る \$_GET['attendee_name'] 参加者名(文字列)
機能	
<p>view_event.php 画面のコントローラーです。画面を制御する各セッションを設置して、view_event.php に移る。</p> <p>view_event.php 画面のフォームを表示するかの制御変数とフォームのアクション処理ファイルを設置する。ゲット方法でクリックした参加者を取得して、input 欄にデフォルト値として入れる。</p>	

4.3.5 モジュール `post_form.php`

入力	セッション変数で次の値を受け取る <code>\$_SESSION['event_id']</code> イベント ID(文字列) <code>\$_SESSION['event_info']</code> イベント情報(リスト) ポスト方法で次の値を受け取る <code>\$_POST['attendeeName']</code> 参加者名(文字列) <code>\$_POST['comment']</code> コメント(文字列)
機能	
view_event.php 画面にあるフォームのアクション処理ファイルです。ユーザー入力を受け取って、データベースに新規参加者情報を insert する。	

4.3.6 モジュール `post_form_modify.php`

入力	セッション変数で次の値を受け取る <code>\$_SESSION['event_id']</code> イベント ID(文字列) <code>\$_SESSION['event_info']</code> イベント情報(リスト) <code>\$_SESSION['modifying_attendee_name']</code> 元の参加者名(文字列) ポスト方法で次の値を受け取る <code>\$_POST['attendeeName']</code> 変更後の参加者名(文字列) <code>\$_POST['comment']</code> コメント(文字列)
機能	
view_event.php 画面にあるフォームのアクション処理ファイルです。ユーザー入力を受け取って、データベースに既存参加者情報を変更する。	