イベントシステム

機能設計書

鈕　皛

2019

内　容

[1. システムの開発環境 2](#_Toc11082216)

[2. 機能概要 2](#_Toc11082217)

[2.1フィールド一覧 2](#_Toc11082218)

[2.2インターフェース一覧 4](#_Toc11082219)

[2.3Controllerのメソッド一覧 7](#_Toc11082220)

[2.4画面一覧 10](#_Toc11082221)

[3.機能設計 12](#_Toc11082222)

[3.1ログイン（FN-01） 12](#_Toc11082223)

[3.2ログアウト（FN-02） 12](#_Toc11082224)

[3.3今日のイベント一覧（FN-03） 12](#_Toc11082225)

[3.4イベント一覧（FN-04、FN-20） 13](#_Toc11082226)

[3.5イベント詳細（FN-05、FN-06、FN-07、FN-21） 13](#_Toc11082227)

[3.6イベント登録（FN-08） 14](#_Toc11082228)

[3.7イベント編集（FN-09） 14](#_Toc11082229)

[3.8イベント削除（FN-10） 15](#_Toc11082230)

[3.9ユーザ一覧（FN-11） 15](#_Toc11082231)

[4.0ユーザ詳細（FN-12） 15](#_Toc11082232)

[4.1ユーザ登録（FN-13） 16](#_Toc11082233)

[4.2ユーザ編集（FN-14） 16](#_Toc11082234)

[4.3ユーザ削除（FN-15） 16](#_Toc11082235)

[4.4自分参加したイベント一覧（FN-16、FN-17、FN-18、FN-19） 17](#_Toc11082236)

[4.5イベント検索（FN-20） 17](#_Toc11082237)

[4.更新履歴 17](#_Toc11082238)

# システムの開発環境

本システムの開発言語、開発環境として以下のものを使用する。

|  |  |
| --- | --- |
| 開発環境 | Tomcat |
| 開発言語 | JAVA（JDK1.8） |
| フレームワーク | Spring+Hibernate |
| データベース | MySQL |
| Webフレームワーク | Bootstrap（v4.3） |

# 機能概要

本システムは、利用者がイベント情報を登録し共有する機能を提供する。

## 2.1フィールド一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| グラス | フィールド名 | 修飾子と型 | 種類 | 説明 |
| User | userId | Private | Integer |  |
| loginId | Private | String |  |
| pass | Private | String |  |
| userName | Private | String |  |
| type | Private | Type |  |
| group | Private | Group |  |
| createdate | Private | Date |  |
| Img | Private | String |  |
| Mail | Private | String |  |
| Type | typeId | Private | Integer |  |
| typeName | Private | String |  |
| ADMIN | Public static final | int |  |
| Join | joinId | Private | Integer |  |
| user | Private | User |  |
| event | Private | Event |  |
| Group | groupId | Private | Integer |  |
| groupName | Private | String |  |
| Event | eventId | Private | Integer |  |
| title | Private | String |  |
| startdate | Private | Date |  |
| enddate | Private | Date |  |
| place | Private | String |  |
| group | Private | Group |  |
| details | Private | String |  |
| user | Private | User |  |
| createdate | Private | Date |  |

## 2.2インターフェース一覧

|  |  |  |  |
| --- | --- | --- | --- |
| グラス | 修飾子と型 | メソッド | 説明 |
| UserDao | List<User> | findAll() | 全てUser情報を取得する |
| User | findById(Integer id) | 指定したidのUser詳細情報を取得する。 |
| void | insert(User user) | Userデータを追加する。 |
| void | update(User user) | Userデータを更新する。 |
| void | delete(User user) | Userデータを削除する。 |
| User | findByLoginIdAndLoginPass(String loginId, String loginPass ) | 指定したUserIDとパスワードとデータベースと比較する。 |
| GroupDao | List<Group> | findAll() | グループ情報を取得する。 |
| JoinDao | List<Join> | findAll() | 全てｊoin情報を取得する |
| Join | findById(Integer id) | 指定したidのｊoin情報を取得する。 |
| void | insert(Join ｊoin) | ｊoinデータを追加する。 |
| void | update(Join ｊoin) | ｊoinデータを更新する。 |
| void | delete(Join ｊoin) | ｊoinデータを削除する。 |
| Join | findByUserAndEvent(User user, Event event) | 指定したｊoinID情報のUserIDとEventIDを取得する。 |
| List<Join> | findByEvent(Event event) | 指定したｊoinID情報のEventIDを取得する。 |
| TypeDao | List<Type> | findAll() | ユーザ身分の情報を取得する。 |
| EventDao | List<Event> | findAll() | 全てEvent情報を取得する |
| Event | findById(Integer id) | 指定したidのEvent詳細情報を取得する。 |
| void | insert(Event event) | Eventデータを追加する。 |
| void | update(Event event) | Eventデータを更新する。 |
| void | delete(Event event) | Eventデータを削除する。 |
| Event | findByLoginIdAndLoginPass(String loginId, String loginPass ) | 指定したUserIDとパスワードとデータベースと比較する。 |
| BaseDao | Session | getSession() | sessionFactoryの情報を取得する。 |
| BaseDaoOfMailSender | void | send(SimpleMailMessage mail) | SimpleMailMessageの情報を取得する。 |

## 2.3Controllerのメソッド一覧

|  |  |  |
| --- | --- | --- |
| グラス | メソッド | 説明 |
| AuthController | public String addGet(Model model) |  |
| public String loginPost(@Valid User user,Errors errors,HttpSession session) |  |
| public String logout(HttpSession session) |  |
| EventController | public void InitBinder(WebDataBinder binder) |  |
| public String list(Model model, HttpServletRequest request) | イベント一覧 |
| public String addGet(Model model, HttpSession session) | イベント情報登録 |
| public String addPost(@Valid Event event,Errors errors,Model model) |  |
| public String editGet(@PathVariable("id") Integer eventId, Model model, HttpSession session) | イベント編集 |
| public String editPost(@Valid Event event, Errors errors, Model model, HttpSession session) |  |
| public String todaylist(@RequestParam(name = "p", defaultValue = "1") String page, Model model) | 本日のイベント |
| public String detail(@PathVariable("id") Integer eventId, Model model, HttpSession session) | イベント詳細 |
| public String del(@PathVariable("id") Integer id, Model model, HttpSession session) | イベント詳細削除 |
| public String　myCreatedList(@RequestParam(value = "p", defaultValue = "0") String p, Model model, HttpSession session, HttpServletRequest request) | 該当ユーザ登録したイベント情報を取得する |
| public String myJoinList(@RequestParam(value = "p", defaultValue = "0") String p, Model model, HttpSession session, HttpServletRequest request) | 該当ユーザ参加したイベント情報を取得する |
| JoinController | public String joinEvent(@PathVariable("id") Integer eventId, Model model, HttpSession session) | イベントに参加する |
| public String cancelEvent(@PathVariable("id") Integer eventId, Model model, HttpSession session) | イベントの参加を取り消す |
| UserController | public String userlist(@RequestParam(name = "p", defaultValue = "1") String page, Model model) | ユーザー一覧(管理者) |
| public String addGet(Model model, HttpSession session) | ユーザー登録 |
| public String addPost(@Valid User user, Errors errors, Model model) |  |
| public String detail(@PathVariable("id") Integer userId, Model model) | ユーザー詳細 |
| public String del(@PathVariable("id") Integer id, Model model, HttpSession session) | ユーザー削除 |
| public String editGet(@PathVariable("id") Integer userId, Model model, HttpSession session) | ユーザー編集 |
| public String editPost(@RequestParam("passNew") String passNew, @Valid User user, Errors errors, HttpSession session, Model model) |  |

## 2.4画面一覧

|  |  |  |  |
| --- | --- | --- | --- |
| 画面ID | URL | Controller | View |
| PG-01 | /login | AuthController | Login.jsp |
| PG-02 | /logoutDone | AuthController | logoutDone.jsp |
| PG-03 | /todayEvent | EventController | todayEvent.jsp |
| PG-04 | /eventList | EventController | eventList.jsp |
| PG-05 | /detailsEvent | EventController  /JoinController | detailsEvent.jsp |
| PG-06 | /addEvent | EventController | addEvent.jsp |
| PG-07 | /addEventDone | EventController | addEventDone.jsp |
| PG-08 | /editEvent | EventController | editEvent.jsp |
| PG-09 | /editEventDone | EventController | editEventDone.jsp |
| PG-10 | /editEvent | EventController | editEvent.jsp |
| PG-11 | /delEvent | EventController | delEvent.jsp |
| PG-12 | /userList | UserController | userList.jsp |
| PG-13 | /detailsUser | UserController | detailsUser.jsp |
| PG-14 | /addUser | UserController | addUser.jsp |
| PG-15 | /addUserDone | UserController | addUserDone.jsp |
| PG-16 | /editUser | UserController | editUser.jsp |
| PG-17 | /editUserDone | UserController | editUserDone.jsp |
| PG-18 | /editUser | UserController | editUser.jsp |
| PG-19 | /delUser | UserController | delUser.jsp |
| PG-20 | /myEvent | MyController | myEvent.jsp |
| PG-21 | /login | AuthController | Login.jsp |

# 3.機能設計

## 3.1ログイン（FN-01）

* 入力しない状態で、「次へ」のボタン押すとErrorを表示する。
* UserIDを入力し、データベースの中で存在の場合は、次のパスワード画面に遷移する。データベースの中で存在しないの場合は、Errorを表示する。
* パスワードを入力し、正しい場合は、Event一覧画面に遷移する。間違い場合はErrorを表示する。

## 3.2ログアウト（FN-02）

* 任意の画面で「ログアウト」Linkを押すとSessionデータを削除し、ログイン画面に遷移する。

## 3.3今日のイベント一覧（FN-03）

* 今行いイベントを取得し、リストで表示する。
* 毎ページ取得したイベント情報を5個記録で表示する。
* イベント情報の「詳細」ボタン押すと、イベント詳細画面に遷移する。

## 3.4イベント一覧（FN-04、FN-20）

* 全てイベントを取得し、最新日時の順位にリストで表示する。
* 毎ページ取得したイベント情報を5個記録で表示する。
* イベント情報の「詳細」ボタン押すと、イベント詳細画面に遷移する。
* 「イベント登録」ボタン押すと、イベント登録画面に遷移する。
* 参加したのイベントはタイトルの右に「参加」のマックを表示する。
* 検索ワードを入力し、全てそのワードに関するイベント情報が表示する。

## 3.5イベント詳細（FN-05、FN-06、FN-07、FN-21）

* 指定したIDのイベント情報を取得する。
* 参加したのイベントはタイトルの右に「参加」のマックを表示する。
* 「イベント一覧に戻る」ボタン押すとイベント一覧画面に遷移する。
* 「参加する」ボタン押すとイベントはタイトルの右に「参加」のマックを表示し、イベントの管理者にメールを送信し、「参加を取り消す」を表示する。
* 「参加を取り消す」ボタン押すとイベントはタイトルの右に「参加」のマックを消して、イベントの管理者にメールを送信し、「参加する」を表示する。
* 管理ユーザの場合は、全ての（自分登録したと他の人登録した）イベントを編集、削除できる。
* 普通ユーザの場合は、自分登録したイベントのみ編集、削除できる。
* 「編集」ボタン押すとイベント編集画面に遷移する。
* 「削除」ボタン押すとイベント削除提示画面が表示する。「OK」を押すとイベント削除画面に遷移する。

## 3.6イベント登録（FN-08）

* タイトル、開始日時、場所を入力しない状態で「登録」ボタン押すと、Errorを表示する。
* 対象グロープの選択項目「全員」が表示する。
* 「キャンセル」ボタン押すとイベント一覧画面に遷移する。
* 「登録」ボタン押すとイベント登録完了画面に遷移する。

## 3.7イベント編集（FN-09）

* タイトル、開始日時、場所を入力しない状態で「登録」ボタン押すと、Errorを表示する。
* 対象グロープの選択項目「全員」が表示する。
* 「キャンセル」ボタン押すとイベント詳細画面に遷移する。
* 「保存」ボタン押すとイベント編集完了画面に遷移する。

## 3.8イベント削除（FN-10）

* イベント詳細画面で「削除」ボタン押すと削除提示aerltを表示し、「OK」を押すとイベントを削除しましたの画面に遷移する。

## 3.9ユーザ一覧（FN-11）

* 普通ユーザの場合は、ユーザ詳細画面に遷移すると、自動的にイベント一覧画面に遷移する。
* 全てユーザを取得し、UserIDの順位にリストで表示する。
* 毎ページ取得したユーザ情報を5個記録で表示する。
* ユーザ情報の「詳細」ボタン押すと、ユーザ詳細画面に遷移する。
* 「ユーザ登録」ボタン押すと、ユーザ登録画面に遷移する。

## 4.0ユーザ詳細（FN-12）

* 指定したIDのユーザ情報を取得する。
* 「ユーザ一覧に戻る」ボタン押すとユーザ一覧画面に遷移する。
* 管理ユーザの場合は、全てのユーザを編集、削除できる。
* 普通ユーザの場合は、ユーザ詳細画面に遷移すると、自動的にイベント一覧画面に遷移する。
* 「編集」ボタン押すとユーザ編集画面に遷移する。
* 「削除」ボタン押すとユーザ削除提示画面が表示する。「OK」を押すとイベント削除画面に遷移する。
* 登録したUser情報自分のUser情報を削除できない。

## 4.1ユーザ登録（FN-13）

* 全部項目を入力しない状態で「登録」ボタン押すと、Errorを表示する。
* 「キャンセル」ボタン押すとユーザ一覧画面に遷移する。
* 「保存」ボタン押すとユーザ登録完了画面に遷移する。
* 写真を追加しない場合は、「default.jpg」写真を追加する。

## 4.2ユーザ編集（FN-14）

* 全部項目を入力しない状態で「登録」ボタン押すと、Errorを表示する。
* 「キャンセル」ボタン押すとユーザ詳細画面に遷移する。
* 「保存」ボタン押すとユーザ編集完了画面に遷移する。
* 写真を修正する場合は、新しい写真を表示する。

## 4.3ユーザ削除（FN-15）

* ユーザ詳細画面で「削除」ボタン押すと削除提示aerltを表示し、「OK」を押すとユーザを削除しましたの画面に遷移する。

## 4.4自分参加したイベント一覧（FN-16、FN-17、FN-18、FN-19）

* 全て自分登録したイベントと参加したイベント情報を取得し、日時の順位にリストで表示する。
* 毎ページ取得したイベント情報を5個記録で表示する。
* イベント情報の「詳細」ボタン押すと、イベント詳細画面に遷移する。
* 「イベント登録」ボタン押すと、イベント登録画面に遷移する。

## 4.5イベント検索（FN-20）

* 毎ページ取得したイベント情報を5個記録で表示する。

# 4.更新履歴

|  |  |  |
| --- | --- | --- |
| 日付 | 編集者 | 内容 |
| 2019年6月10日 | 鈕皛 | 第1.0 版作成 |
| 2019年6月18日 | 鈕皛 | 第1.1 版作成 |