|  |  |  |
| --- | --- | --- |
| TASK（仮） | | |
|  |  |  |
| 要件定義・設計書Ver. 1.0 | | |

# 変更履歴

|  |  |  |
| --- | --- | --- |
| Version | 更新日付 | 更新内容 |
| 1.0 | 2020/3/13 | 初版作成 |

# アプリの目的

**1.** クエストを作成してタスクを管理できるアプリ

**2.** 一つの予定でも細かくタスクを分けることで進めやすくなる

**3.**

# 主な機能要件

ユーザーについて、クエストについてなど分けて書くべき

1. ユーザーは一人ずつアカウントを持ち、ログインすることでクエストの作成、編集、確認が可能となる。
2. 受注中のクエスト一覧が表示される
3. ユーザーは自分でクエストを作成し、タイトル、達成条件、期限などを設定する。
4. 協力プレイも可能、クエストを複数人で共有し、達成したものは達成した人がチェックをつける、タスクに手を付けるときに作業中のフラッグを立てて周りに知らせる。
5. クエストにはその内容と、予定の作成者、編集者、作成日時、共有するユーザーまたはチームが記載される。
6. クエストひとつごと、チームひとつごとでチャットができる。
7. 画像を添付すればクエストのサムネイルを設定できる。

# 機能要件の詳細（赤字は未実装）

1. **ログイン**

・デフォルトページにログイン情報、ログイン、新規登録のボタンを設置

・ユーザーIDまたはメールアドレスとパスワードでユーザー認証を行う。

・認証されればメイン画面に遷移する。

・ユーザーはまず登録されていなけらばならない他のページにアクセスしてもデフォルトページに遷移される。

1. **ログアウト**

メイン画面のログアウトボタンからログアウトしてデフォルトページに戻る、

1. **ユーザー新規登録**

・ユーザーID、メールアドレス、パスワード、ユーザー名を登録してアカウントを作成する。

・既に登録されているユーザーID、メールアドレスでアカウントは作成できない。

1. **クエストを表示**

自分が受注しているクエストがデスク（メイン画面）に並べて表示される

掲示板には参加しているボード（チーム）のクエストすべてが表示される

クエストの内容は

・予定識別ID

・期限日時と残り時間

・タイトル

・詳細

・クリア済みかどうか

・作成したユーザーのID

・参加しているユーザー

・参加しているチーム

・チャット

・チェックリスト

1. クエストの受注(take)
2. クエストの作成(post)・編集・削除
3. チームの表示
4. チームの作成
5. ごみの日

# テーブル設計

1. userテーブル（赤字は未実装）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **型（桁）** | **制約** | **備考** |
| user\_id | VARCHAR(15) | PRIMARY KEY | ユーザーID |
| pass | VARCHAR(20) | NOT NULL | パスワード |
| mail | VARCHAR(255) | NOT NULL | メールアドレス |
| user\_name | VARCHAR(30) | NOT NULL | ユーザー名 |
| created\_at | DATETIME | NOT NULL | アカウント作成日時 |
| quest\_ids\_of\_user |  |  |  |

‘yuuki’,’pass’,’mail’,’yuukiname’,’2020-03-13 18:55:00’（仮ユーザー）

ユーザー検索時にはuser\_idはbyte型の方が検索が速くていいのでは

1. questテーブル

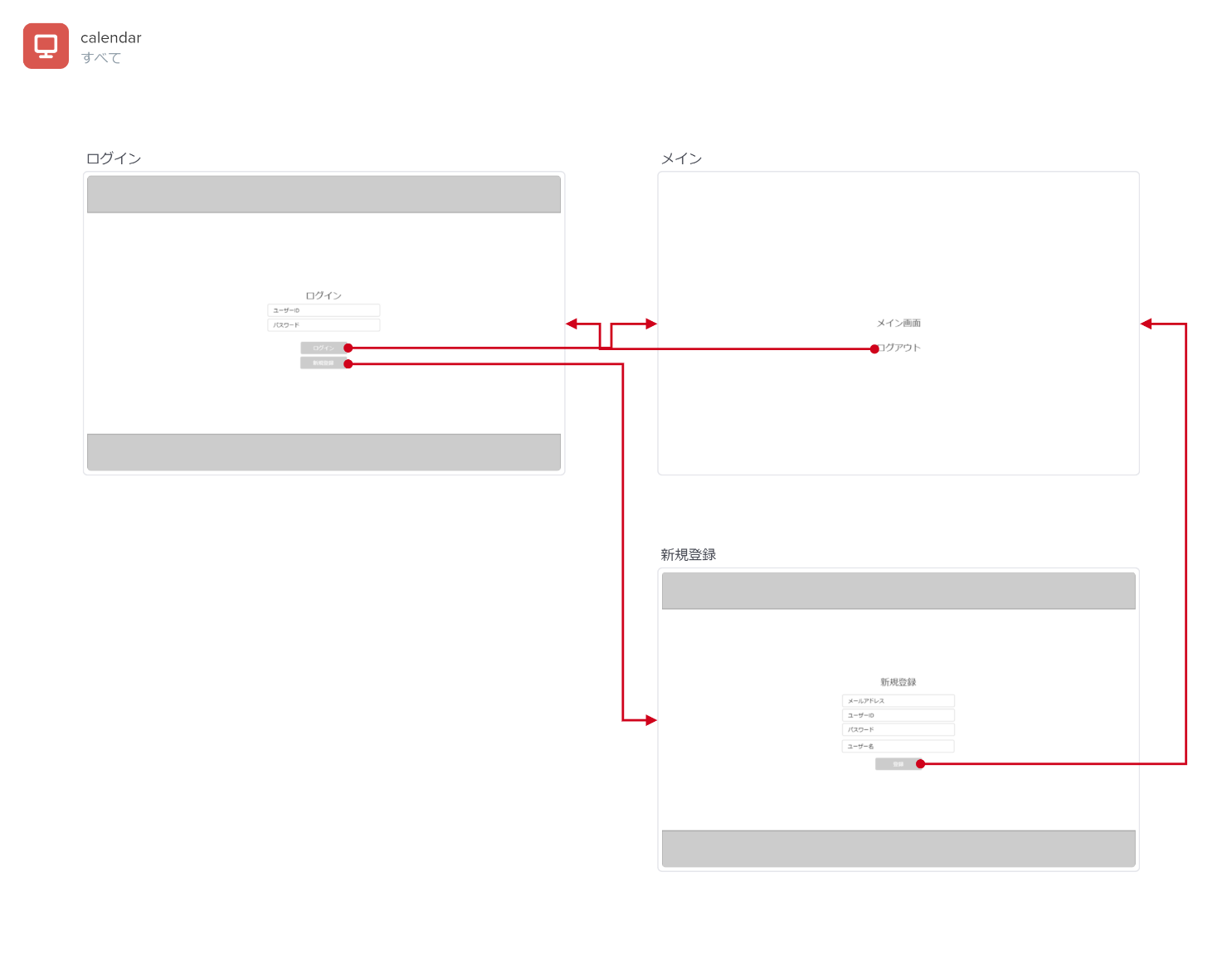
|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **型（桁）** | **制約** | **備考** |
| quest\_id | INT | PRIMARY KEY  AUTO INCREMENT | 予定識別ID |
| title | VARCHAR(30) | NOT NULL | クエストのタイトル |
| details | VARCHAR(10000) |  | クエストの詳細 |
| deadline | DATETIME |  | 期限 |
| cleared | BOOLEAN | NOT NULL | クリア済みかどうか |
| joined\_users | USER | NOT NULL | クエストに参加しているユーザー |
| team | team | NOT NULL | 参加しているチーム |
| chat |  |  | チャット |
| check\_list |  |  | チェックリスト |
| created\_user\_id |  |  | 予定を作成したユーザーのID |
| created\_at | Timestamp | NOT NULL | 予定作成日時 |

1. takes

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **型（桁）** | **制約** | **備考** |
| id | INT(20) | NOT NULL PRIMARY KEY  AUTO INCREMENT |  |
| take\_id | INT | NOT NULL | テイクされたクエストのID |
| user\_id | VARCHAR(15) | NOT NULL | ユーザーID |
| created\_at | DATETIME | NOT NULL | テイク日時 |

# 機能の画面設計

1. ログイン、新規登録、ログアウト



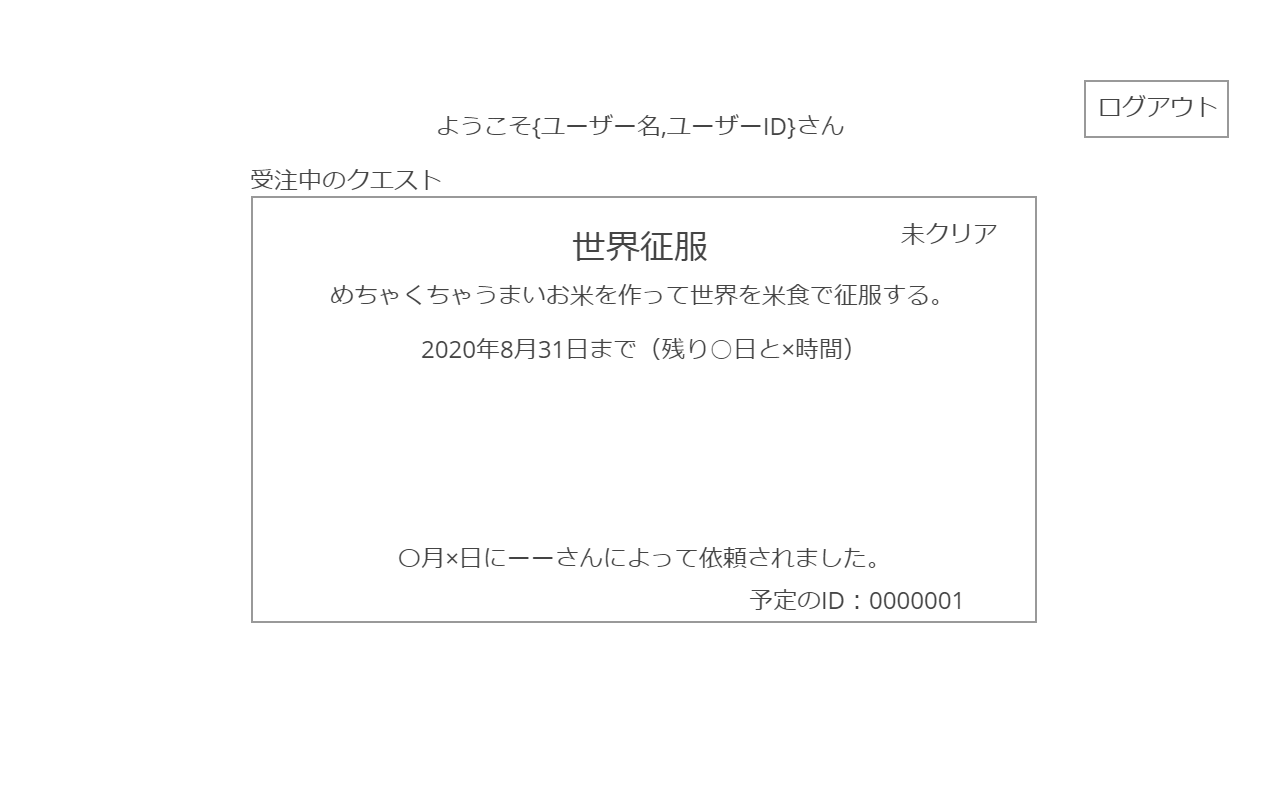
GET LogoutServlet フォワード→index.jsp

index.jsp

POST RegisterServlet フォワード→main.jsp

GET RegisterServlet フォワード→register.jsp

POST LoginServlet フォワード→main.jsp

1. クエストの表示（メイン画面）

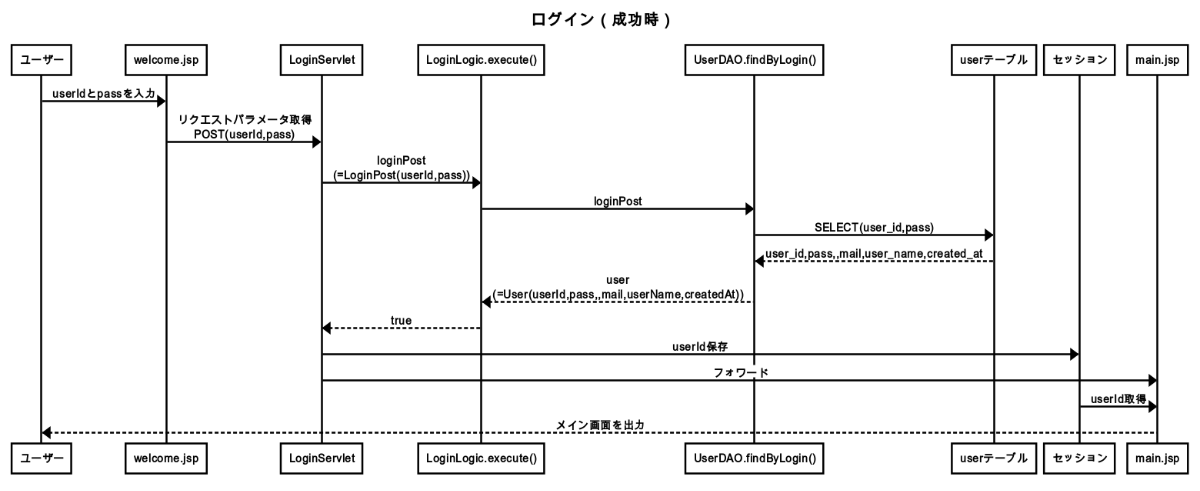
GET MainServlet フォワード→main.jsp

1. クエストの作成

クエストを作成し、掲示板に貼り出す。

1. クエストの受注（take）

# サーブレット・JSPの設計

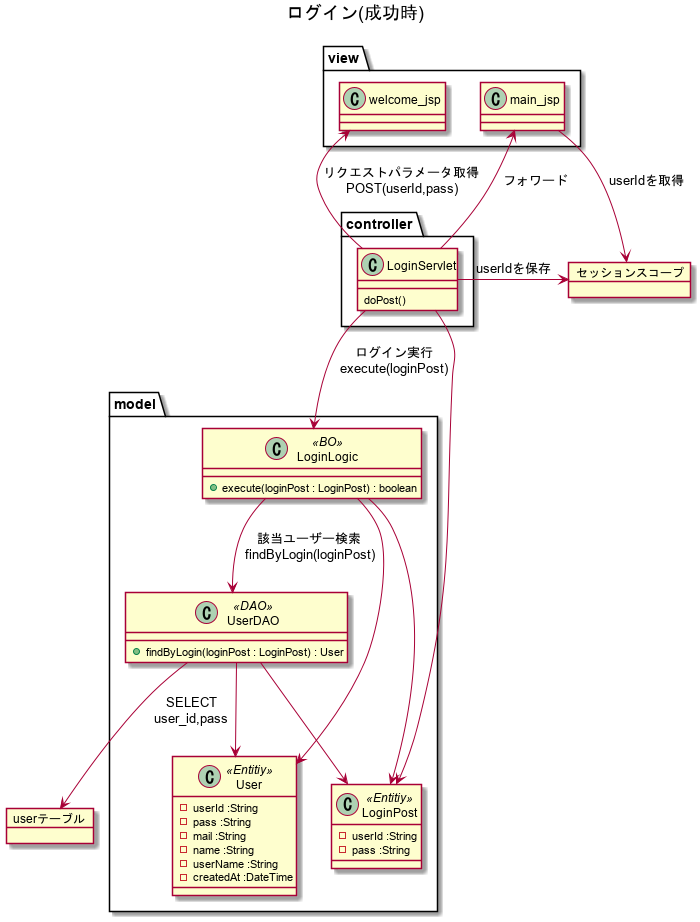
1. ログイン

------- user

userNameも保存

------- index.jsp

------- LoginInfo



userNameも保存

------- User

-------

index\_jsp

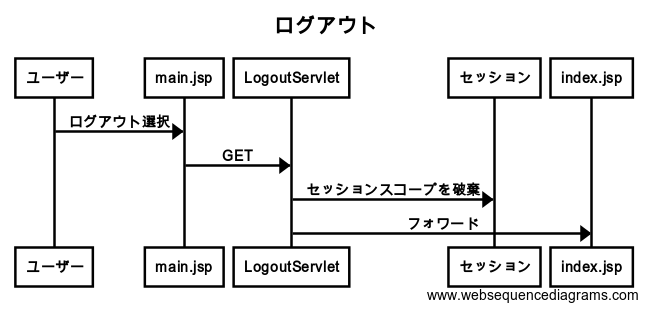
------- jsp

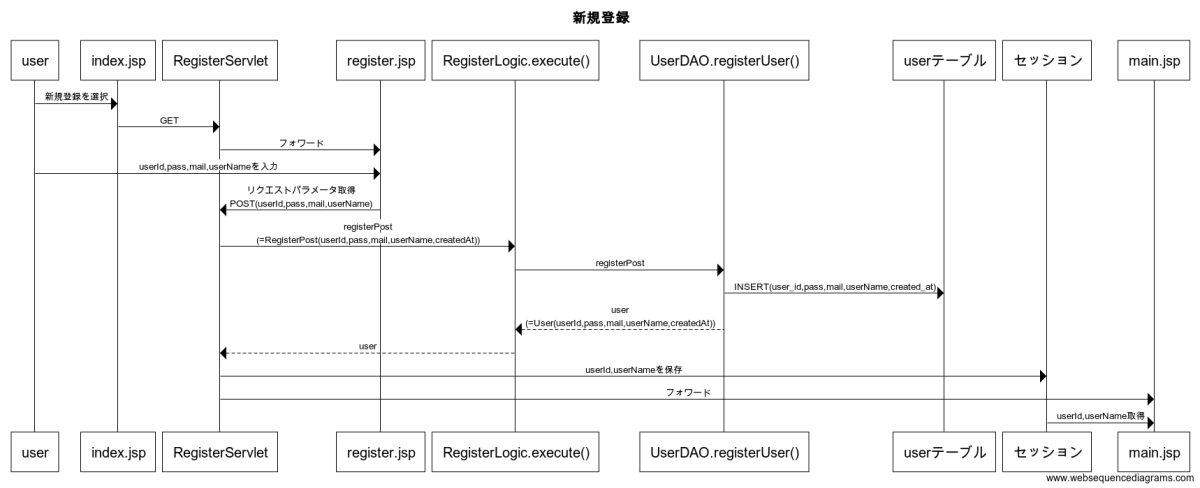
------- servlet

------- LoginInfo

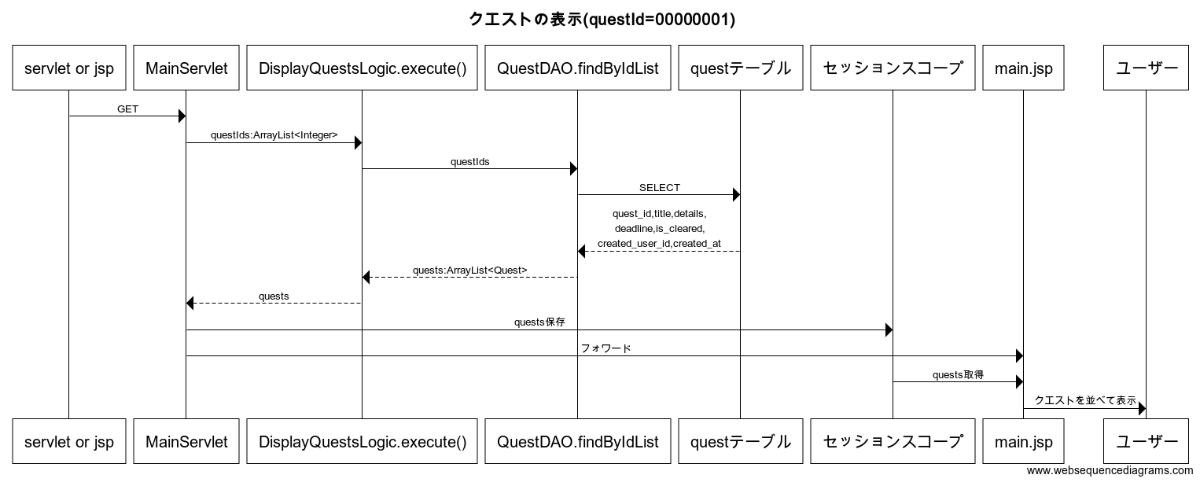
DAOはインターフェース作る

1. ログアウト

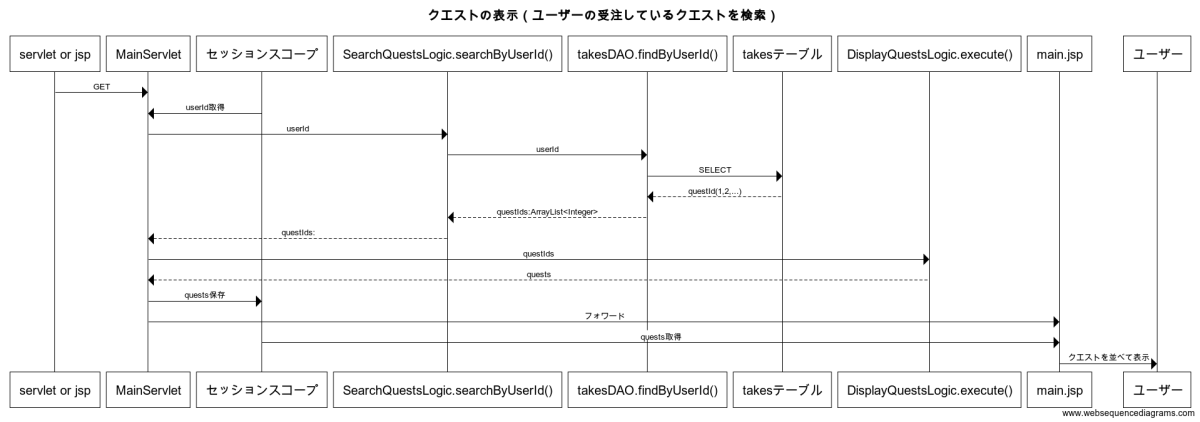


1. 新規登録

POSTでの文字コード指定はCharacterCodeFilterでフィルターをかけて行う

1. クエストの表示。questId=00000001（リストのクエストIDで指定されたクエストを表示するだけ。）

--

1. クエストの表示。ユーザーが受注しているクエストを検索し、それらのクエストIDでリストを生成する。ログイン中のユーザーが受注しているクエストだけを表示する。

アプリ作成のタスク管理をしているアプリ↓Trello





# 必要なもの