

簡易ロードマップ作成アプリ(仮)

基礎情報

作成期日

要件

目的

概要と作成理由

ターゲット(ペルソナ)

開発環境・使用言語

アプリケーション機能

1. ログイン・ログアウト機能
2. アカウント管理機能(作成・変更・削除)
3. ロードマップ作成・保存機能
4. ロードマップ管理機能(変更)
5. ロードマップ管理機能(全削除・全コピー)

※リリース後の改善予定

クラス設計

テーブル設計

基礎情報

作成期日

2024/12/24(火)

要件

目的

Webアプリケーション「簡易ロードマップ作成アプリ(仮)」の開発。

概要と作成理由

メイン言語が日本語かつ簡単にシンプルなロードマップを作成できるWebアプリケーション。

キャリアパスとしてのロードマップを作成できる無料のWebアプリケーションである「

[roadmap.sh](#)」に関して、日本語特化した類似のものが無い点、ロードマップ作成

のカスタマイズ性が豊富で、作成にそれなりに時間が必要となる点から、メイン言語が日本語かつシンプルにタイパ重視で作成できるようなものがあれば、英語に不慣れな初学者でも簡単に利用できると感じたため。

ターゲット(ペルソナ)

母国語が日本語、かつ初学者など自身のロードマップを簡単に作成したいと考えるユーザー。

開発環境・使用言語

開発環境：

Windows10、Eclipse2023、Apache Tomcat(Tomcat10(Java21))、H2データベース

使用言語：

Java (Servlet, JSP)、HTML5、CSS5、JavaScript

アプリケーション機能

1. ログイン・ログアウト機能

ユーザーごとにロードマップを管理するためのログイン機能。

ユーザーID、パスワードを用いたログイン機能。パスワードを忘れた場合はメールアドレスを入力することで、パスワード再設定のリンクを送信する。

2. アカウント管理機能(作成・変更・削除)

ユーザーを管理するアカウントを作成する機能に加え、登録情報(ID、パスワード、メールアドレス)の変更機能、アカウントを削除する機能を有する。

3. ロードマップ作成・保存機能

ロードマップのデザインは固定。ユーザーは入力フォーム形式で情報を入力することでロードマップを作成できる。

フォーム部分での入力内容は以下の通り。なお、要素番号は連番となるように必要に応じて内部で再採番する。

- ・ロードマップの名称 *空白の場合はHTMLでエラー表示
- ・親要素作成順 *空白の場合は0が自動入力。同列の親要素がある場合は処理順。

・親要素名	※空白の場合は、子要素の有無に関わらず作成しない。
・子要素作成順	※空白の場合は0が自動入力。同列の子要素がある場合は処理順。
・子要素名	※空白の場合は作成しない。
・子要素のタグ(選択なし、必須、選択、余力があれば)	※デフォルトは「選択なし」。

必要に応じてユーザーが親要素名・子要素名の入力欄を追加でき、要素の数を増減できる。

作成したロードマップはユーザーごとに保存する機能を有する。

4. ロードマップ管理機能(変更)

ユーザーごとに保存したロードマップに関して、追記・部分削除・内容変更が可能な変更機能、ロードマップそのものを削除する削除機能、ロードマップをコピーして他のロードマップとするコピー機能を有する。

なお、ロードマップ作成直後は変更画面に遷移し、要素をクリックすることでモーダルウィンドウが開き、以下の内容を変更可能。

・親要素番号	※ロードマップ開始点からの番号(連番)。 ※挿入箇所指定で順番変更可能(挿入、削除で連番を維持)。
・親要素名	
・子要素番号	※親要素内での番号(連番)。 ※挿入箇所指定で順番変更可能(挿入、削除で連番を維持)。
・子要素名	
・子要素のタグ(選択なし、必須、選択、余力があれば)	
・ステータス	※完了or未完了。完了状態でロードマップ上の要素が薄くなる。
・要素の内容	※各要素の詳細な説明文。

また、変更画面の要素削除ボタンで該当要素を削除可能。

他に、親要素追加ボタン、子要素追加ボタンで該当箇所「0：先頭、1(要素番号)：xx(要素名)の下」のように要素の挿入が可能。

5. ロードマップ管理機能(全削除・全コピー)

ロードマップ変更画面にて、ロードマップ単位で削除、または別のロードマップとしてコピーできる。

※リリース後の改善予定

- ロードマップ変更画面において、親要素をコピーして別の親要素として挿入する機能
- ロードマップの公開設定(リンク共有、ログイン非保持でもアクセス可能)機能

クラス設計

[アプリケーション名]

simpleRoadmapCreator(仮)

【モデル】

[エンティティ]

Login.java：ログイン時の入力情報を格納

Roadmap.java：ユーザーIDに紐づくロードマップを格納

[エンティティ(データベース情報格納するDTO)]

Account.java：ACCOUNTSテーブルの内容を格納

RoadmapId.java：ROADMAPSテーブルの内容を格納

ParentElement.java：PARENT_ELEMENTSテーブルの内容を格納

[BO]

LoginLogic.java：ログイン処理を実施

RoadmapGetLogic.java：ユーザーに紐づくロードマップID取得処理を実施

【DAO】

AccountsDAO.java：Loginに格納されたuserIdとpassで検索するDAO

RoadmapIdsDAO.java：Loginに格納されたuserIdで検索するDAO

ParentElementsDAO：RoadmapIdに格納されたuserIdとroadmapIdで検索するDAO

【サーブレット】

LoginServlet.java：トップ画面(兼ログイン画面)への誘導、ログイン処理を管理

LogoutServlet.java：ログアウト処理を管理

ForgetPassServlet.java：「パスワードを忘れた場合」の処理を管理

PassResetServlet.java：パスワード再設定の処理を管理

RegisterServlet.java：アカウント登録処理を管理

AccountServlet.java：アカウント情報変更・削除処理を管理

MypageServlet.java：マイページ内に関する処理を管理

RoadmapServlet.java：ロードマップに関する処理を管理

【ビュー】

index.jsp：トップ画面(兼ログイン画面)

 *action="login"で遷移してきた場合はエラーメッセージ部分を追記表示

forgetPass.jsp：「パスワードを忘れた場合」の遷移先である、パスワード再設定用リンクを送信するための登録したメールアドレスを入力する画面

forgetPassOK.jsp：「パスワードを忘れた場合」で登録済みメールアドレスの入力を受け付けた場合の画面

accessInvalid.jsp：送付URLアクセスが無効である場合の画面

passReset.jsp：パスワード再設定画面

passResetOK.jsp：パスワード再設定完了画面

register.jsp：アカウント登録画面

registerOK.jsp：アカウント仮登録完了画面

registerAccept.jsp：アカウント本登録完了画面

mypage.jsp：マイページ画面

account.jsp：アカウント管理画面

accountUpdate.jsp：アカウント情報変更確認画面

accountUpdateOK.jsp：アカウント情報変更完了画面

accountDelete.jsp：アカウント削除確認画面

accountDeleteOK.jsp：アカウント削除完了画面

roadmap.jsp：作成したロードマップを単一表示する画面

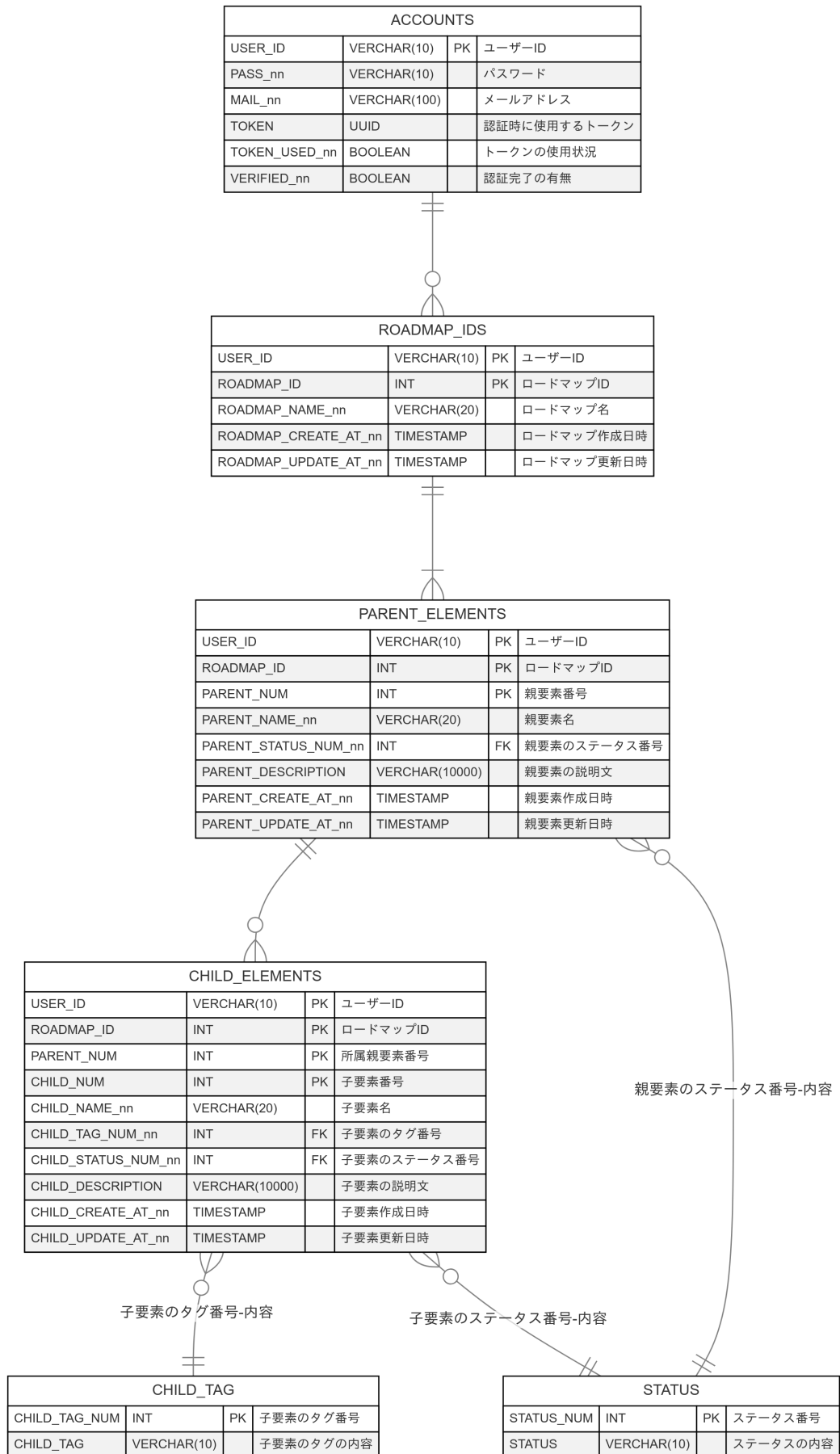
テーブル設計

各テーブルは、列名、データ型、制約、格納情報説明から成る。

※NOT NULL：列名の末尾に「_nn」

PK：主キー

FK：外部キー



%% NOT NULL：列名の末尾に「_nn」

erDiagram

```
ACCOUNTS {
    USER_ID VERCHAR(10) PK "ユーザーID"
    PASS_nn VERCHAR(10) "パスワード"
    MAIL_nn VERCHAR(100) "メールアドレス"
    TOKEN UUID "認証時に使用するトークン"
    TOKEN_USED_nn BOOLEAN "トークンの使用状況"
    VERIFIED_nn BOOLEAN "認証完了の有無"
}

ROADMAP_IDS {
    USER_ID VERCHAR(10) PK "ユーザーID"
    ROADMAP_ID INT PK "ロードマップID"
    ROADMAP_NAME_nn VERCHAR(20) "ロードマップ名"
    ROADMAP_CREATE_AT_nn TIMESTAMP "ロードマップ作成日
時"

    ROADMAP_UPDATE_AT_nn TIMESTAMP "ロードマップ更新日
時"
}

PARENT_ELEMENTS {
    USER_ID VERCHAR(10) PK "ユーザーID"
    ROADMAP_ID INT PK "ロードマップID"
    PARENT_NUM INT PK "親要素番号"
    PARENT_NAME_nn VERCHAR(20) "親要素名"
    PARENT_STATUS_NUM_nn INT FK "親要素のステータス番
号"

    PARENT_DESCRIPTION VERCHAR(10000) "親要素の説明
文"

    PARENT_CREATE_AT_nn TIMESTAMP "親要素作成日時"
    PARENT_UPDATE_AT_nn TIMESTAMP "親要素更新日時"
}

CHILD_ELEMENTS {
    USER_ID VERCHAR(10) PK "ユーザーID"
    ROADMAP_ID INT PK "ロードマップID"
    PARENT_NUM INT PK "所属親要素番号"
    CHILD_NUM INT PK "子要素番号"
    CHILD_NAME_nn VERCHAR(20) "子要素名"
    CHILD_TAG_NUM_nn INT FK "子要素のタグ番号"
```

```

        CHILD_STATUS_NUM_nn INT FK "子要素のステータス番号"
        CHILD_DESCRIPTION VARCHAR(10000) "子要素の説明文"
        CHILD_CREATE_AT_nn TIMESTAMP "子要素作成日時"
        CHILD_UPDATE_AT_nn TIMESTAMP "子要素更新日時"
    }
    STATUS {
        STATUS_NUM INT PK "ステータス番号"
        STATUS VARCHAR(10) "ステータスの内容"
    }
    CHILD_TAG {
        CHILD_TAG_NUM INT PK "子要素のタグ番号"
        CHILD_TAG VARCHAR(10) "子要素のタグの内容"
    }

    ACCOUNTS ||--o{ ROADMAP_IDS : ""
    ROADMAP_IDS ||--|{ PARENT_ELEMENTS : ""
    PARENT_ELEMENTS ||--o{ CHILD_ELEMENTS : ""
    PARENT_ELEMENTS }o--|| STATUS : "親要素のステータス番号-内
容"
    CHILD_ELEMENTS }o--|| CHILD_TAG : "子要素のタグ番号-内容"
    CHILD_ELEMENTS }o--|| STATUS : "子要素のステータス番号-内
容"

```