

2020 年 8 月 11 日 火曜日

仕様書

ペルソナ設定

名前	ken2(自分)
年齢	22 歳
性別	男性
職業	学生(大学院生)
性格	真面目、向上心、努力家
趣味	プログラミング、イラスト、音楽、筋トレ
習慣	PC を毎日使う(プログラミング)
App を使う動機	見やすい bookMarkApp を使いたいため、 PC とスマホで共有するため、 Memo を残すため

ユーザーストーリーマッピング

1. ペルソナの行動

- ・サイトの bookmark を保存
- ・サイトを閲覧
- ・カテゴリー別に bookmark を分ける
- ・なぜ保存したのかを確認/重要な箇所を確認
- ・スマホと PC で共有
- ・いつ何のために保存したのかを確認

2. 欲しい機能

- ・bookmark の新規作成/編集/削除
- ・bookmark の memo 機能
- ・カテゴリー分け
- ・サイトの画像(見やすさ)
- ・検索機能
- ・ログイン/ログアウト(プライバシーの観点から)

3. 機能優先度

- ・ bookmark の新規作成/編集/削除
- ・ bookmark の memo 機能
- ・ カテゴリー分け
- ・ 検索機能
- ・ ログイン/ログアウト(プライバシーの観点から)

4. 実装機能

具体的なものは【制作物の説明】主な機能に記述

Web 版ブックマークアプリ

【課題】

自分が使っているブックマーク(Google Chrome)が、ブックマークしすぎて探しにくくなってしまったこと、またなぜ保存したのかや重要な部分を忘れてしまうこと。

【目的】

見やすい・目的のページを探しやすいブックマーク App を作り、UX の質を上げる、探す時間を省く、メモ機能を追加する。

【目的達成のために作成する物】

Web 版ブックマークアプリ

【制作物の説明】

主な機能 (赤は必須機能)

- ・ URL の保存
 - ・ URL の入力
 - ・ URL の保存
 - ・ タグ付け
 - ・ 日付の追加
 - ・ メモ機能
- ・ URL の編集
 - ・ 編集
- ・ URL の削除

- ・削除

- ・まとめて削除

- ・URL の検索・閲覧

- ・時系列順 or 名前順の表示
- ・ファイルの色付け
- ・日付の表示

- ・検索機能

- ・タグ別閲覧
- ・OGP 機能(取得が難しいので後回し)
- ・階層構造（フォルダーにまとめる）
- ・タグ別グラフ表示

- ・ログイン・アカウント機能

- ・登録（メールアドレス、パスワード）
- ・連携機能
- ・ログイン・ログアウト
- ・プロフィール

【制作物の利用者】

自分

【利用者が得られる便益】

- ・「自分がブックマークしたサイトがどこにあるかわからない」という悩みを解決
- ・探す時間を少なくする
- ・UI、UX の質が向上
- ・メモを残せる
- ・スマホ、PC 間で共有できる

【作るための体制】

個人開発

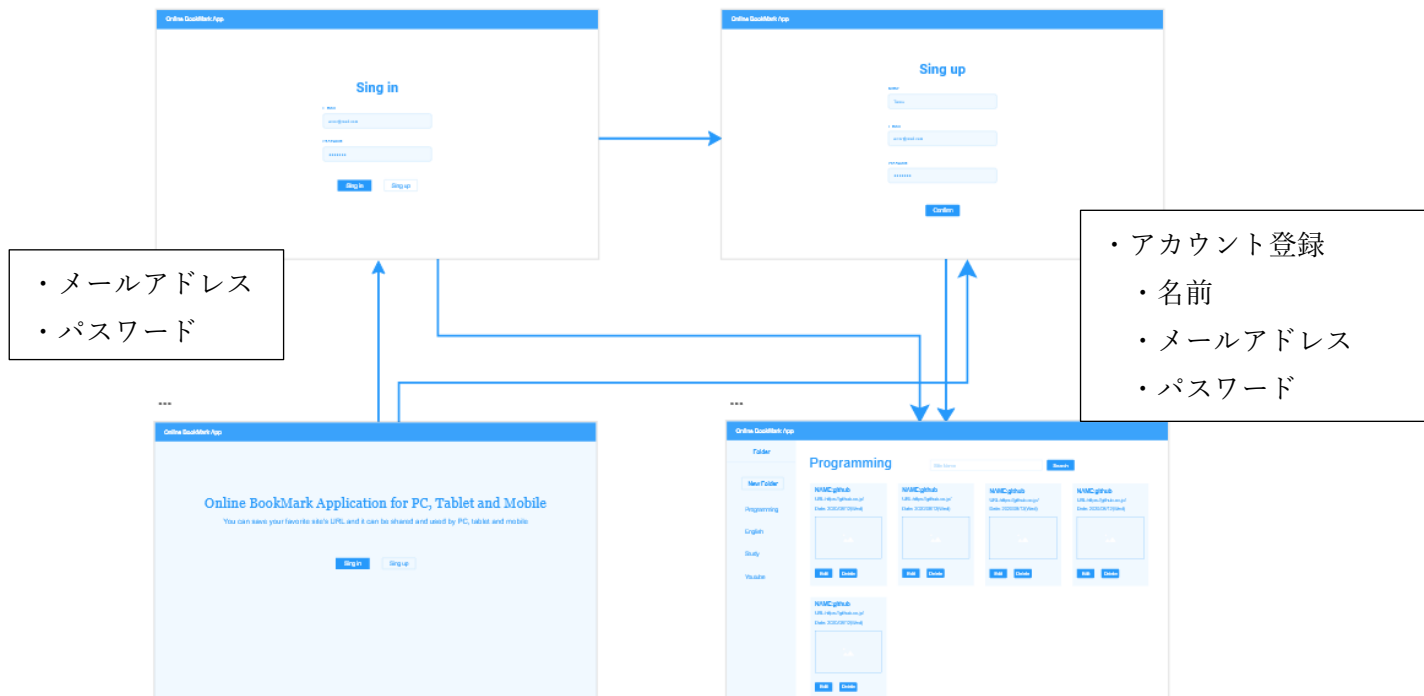
【制作物の期限】

約 1 か月を想定

画面遷移図2020/08/12現在

サインインページ

サインアップページ



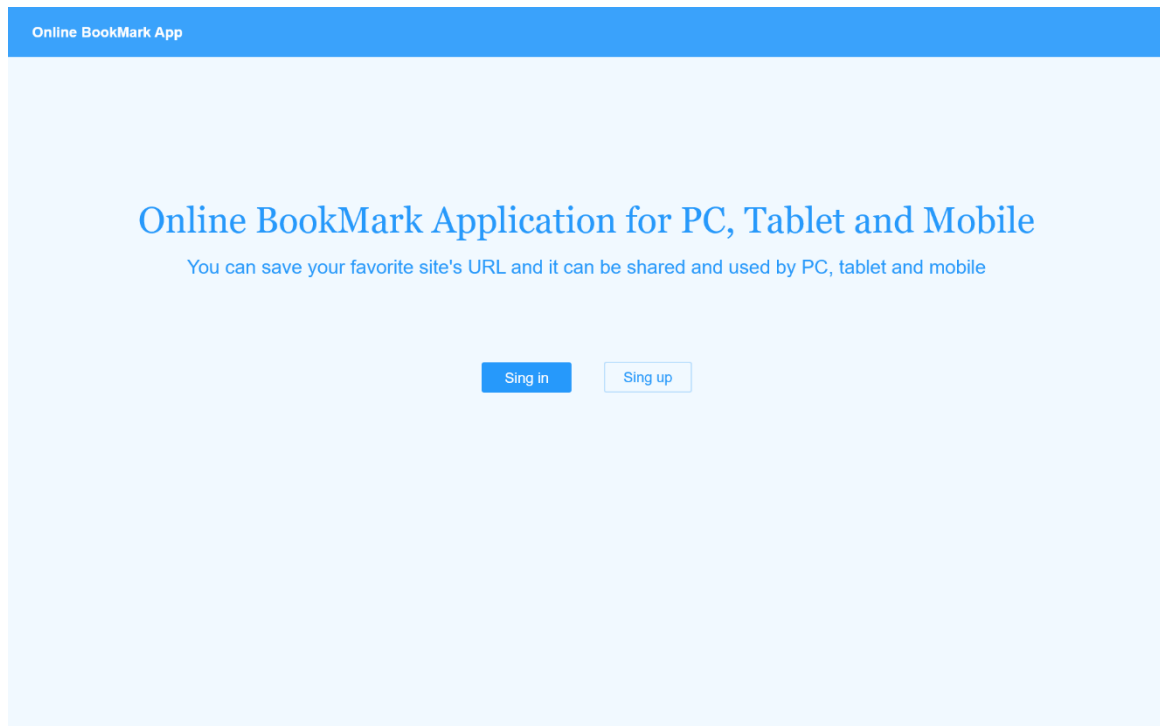
ランディングページ

アカウントページ

- ・ bookmark app の説明
(一目でどういうものかわかる説明)
- ・ サインイン、サインアップへの移動ボタン

- ・ URL のブックマーク
 - ・ 作成
 - サイトの名前
 - URL
 - 作成日時
 - OGP 画像
 - メモ
 - ・ 編集
 - ・ 削除
- ・ フォルダー
 - ・ 作成
 - 色
 - 名前
 - ・ 編集
 - ・ 削除
- ・ 検索機能(絞り込み)
- ・ ログアウト/アカウント切替
- ・ 名前順・日付順に切替

【機能設計】



機能名：ランディングページ
処理内容： <ul style="list-style-type: none">・ アカウントを持っているユーザーにはサインインのページの移動・ アカウントを持っていないユーザーにはサインアップページへ移動
必要なデータ： <ul style="list-style-type: none">・ なし
ユーザー操作： <ul style="list-style-type: none">・ ボタンをクリック

Sing in

E-mail

error@mail.com

Password

Sing in

Sing up

機能名：サインイン

処理内容：

- ・ 入力フォームからメールアドレスとパスワードを受け取る
- ・ 受け取った情報をデータベース(Cloud Firestore)にあるか確認
- ・ 正しいければ、State のサインイン状態を True
- ・ 間違っていれば「メールアドレスもしくはパスワードが間違っています」を表示

必要なデータ：

- ・ 会員情報

取得元

- ・ 入力フォーム
- ・ データベース (Cloud Firestore)

ユーザー操作：

- ・ フォームに入力
- ・ 確認ボタンをクリック

Sing up

Name

Tarou

E-mail

error@mail.com

Password

Confirm

機能名：サインアップ

処理内容：

- ・入力フォームから名前とメールアドレスとパスワードを受け取る、この際に同じメールアドレス and パスワードがないかを確認
- ・確認画面で入力内容を表示
- ・データベース（ユーザー情報ファイル）に保存
- ・State のサインイン状態を True にする

必要なデータ：

- ・会員情報

取得元

- ・入力フォーム

ユーザー操作：

- ・フォームに入力
- ・確認ボタンをクリック
- ・登録ボタンをクリック

Online BookMark App

Folder

New Folder

Programming

English

Study

Youtube

Programming

Site Name

Search

NAME:github
URL:https://github.co.jp/
Date: 2020/08/12(Wed)

EditDelete

NAME:github
URL:https://github.co.jp/
Date: 2020/08/12(Wed)

EditDelete

NAME:github
URL:https://github.co.jp/
Date: 2020/08/12(Wed)

EditDelete

NAME:github
URL:https://github.co.jp/
Date: 2020/08/12(Wed)

EditDelete

NAME:github
URL:https://github.co.jp/
Date: 2020/08/12(Wed)

EditDelete

機能名：ブックマーク登録機能

処理内容：

新規作成

- 入力フォームからサイトの名前(自由に決めれる)と URL と日付、メモ内容を受け取る
- 入力内容された URL をデータベースに保存（非同期処理）
- State に情報を保存
- 情報をもとに View に描画、URL 情報を元に OGP 画像を取得
 - Google API を用いた Favicon を取得に変更

編集機能

- 情報を State から取得
- 後は新規作成の時と同じ

削除

- 確認ダイアログを表示
- データベース、State から削除

必要なデータ：

- 会員情報

取得元

- 入力フォーム

ユーザー操作：

- 新規作成をクリック
- フォームに入力
- 追加をクリック

- 編集をクリック
- フォームに入力
- 修了をクリック

- 削除をクリック
- 確認ダイアログで OK を押す

機能名：フォルダー作成機能
処理内容： <ul style="list-style-type: none">・新規作成をクリック・フォルダー名を入力・作成をクリック
必要なデータ： <ul style="list-style-type: none">・会員情報 取得元 <ul style="list-style-type: none">・入力フォーム・データベース（Cloud Firestore）
ユーザー操作： <ul style="list-style-type: none">・フォームに入力・確認ボタンをクリック

データベース設計

・ユーザー情報

- ユーザーid
- 名前
- E メール
- パスワード
- シリアルフォルダー番号(Folder ID を付与する際に必要になる通し番号)
- シリアル番号(BookMark ID を付与する際に必要になる通し番号)

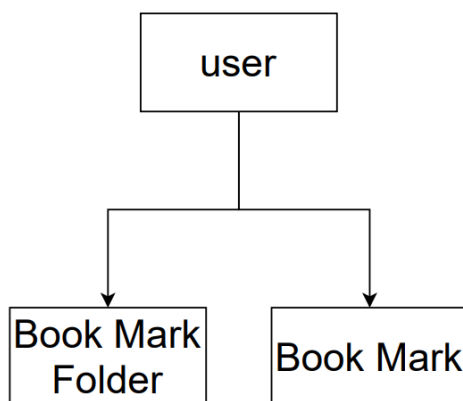
・ブックマークフォルダー

- ユーザーid
- フォルダーid
- フォルダー名
- フォルダー色

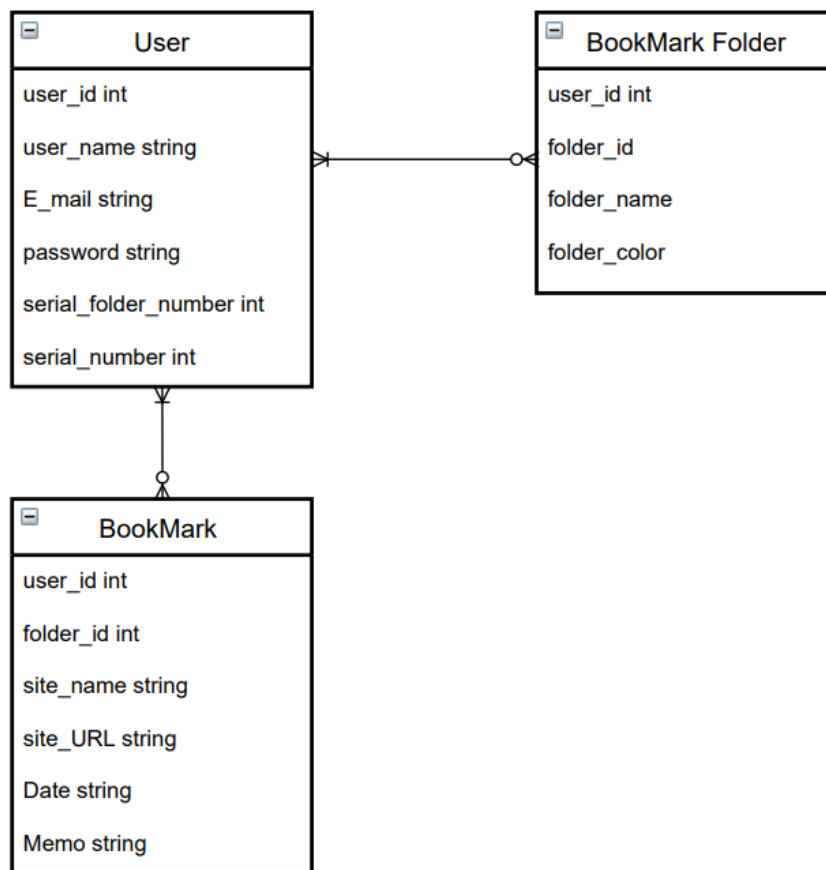
・ブックマーク

- フォルダーid
- ブックマーク id
- サイト名
- URL
- 日付
- メモ

データベースの参照関係

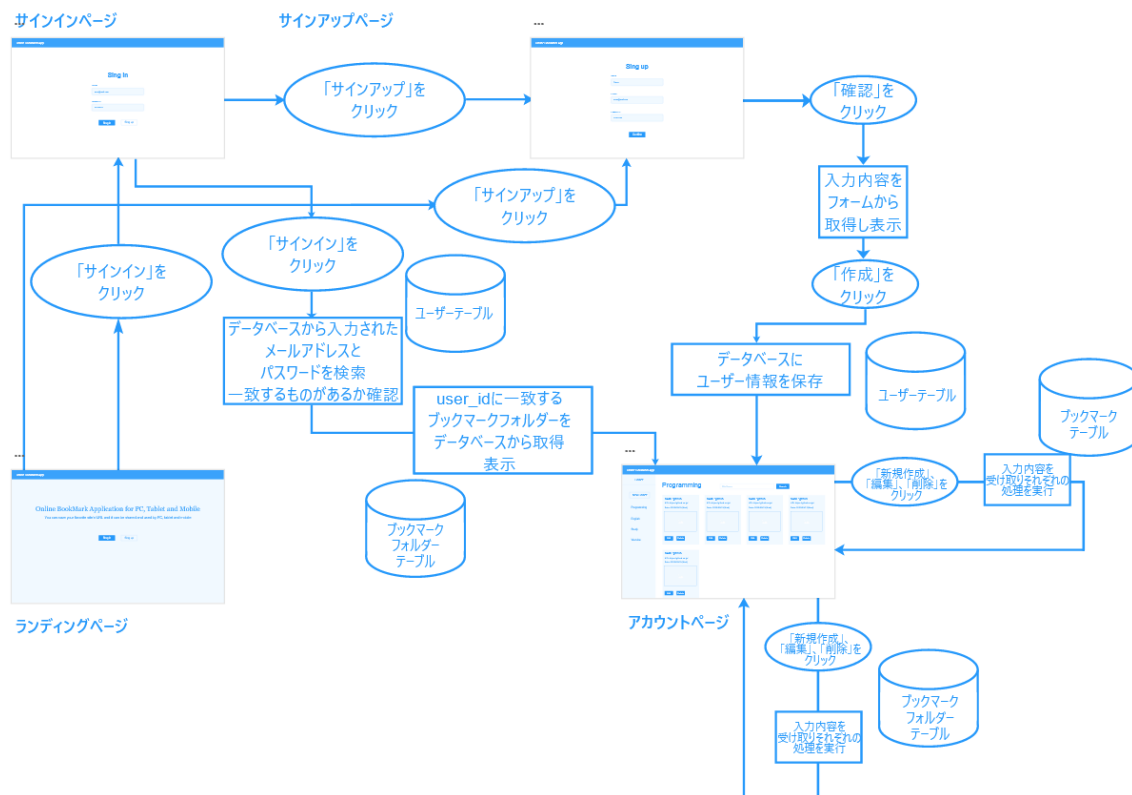


ER 図



データフロー

データフロー図



使用技術：

ホスティング

- ・ Vercel (Hosting)

バージョン管理

- ・ Git/Github

フロントエンド（言語・ライブラリなど）

- ・ TypeScript
- ・ React Hook
- ・ Redux Toolkit
- ・ Material ui
- ・ NextJs

バックエンド

- ・ Firebase(Cloud Function)

DB

- ・ NoSQL(Cloud Firestore)