

概要

開発システム名
Acial Diary

システム概要
AthleteAgent-TECH事業部の社員限定の日記&Todoアプリ Instagramのように投稿できて、またTodoアプリとしても使える

自作開発 アイデア表

1-1. 好きなサービスを出す・よく使ってるサービス

- ・YouTube
- ・Instagram
- ・LINE
- ・note
- ・ZOZOTOWN
- ・Gmail
- ・twitter

1-2. 作ってみたいサービスor既存のサービス

- ・嫌いな人を敵に設定してシューティングしていく爽快FPS
- ・社内限定の日記帳&Todoアプリ(公開したら全員が見れる、非公開は見れない)
- ・社内Instagram

2. ベースはどういう作りか？

- ・YouTube→動画投稿機能
- ・Instagram→投稿機能
- ・LINE→メール機能
- ・note→投稿機能
- ・ZOZOTOWN→ECサイト
- ・Gmail→メール機能
- ・twitter→投稿機能

- ・嫌いな人を敵に設定してシューティングしていく爽快FPS→ゲーム
- ・社内限定の日記帳&Todoアプリ→投稿機能
- ・社内Instagram→投稿機能

3. アイデアを出す(趣味など)

- ・写真を撮ったら内訳の内容を判断してタグ分けする家計簿アプリ
- ・社内の人間だけの連絡ツール(電話もあり)
- ・パーソナルカラー、骨格診断してくれるアプリ
- ・タイピングアプリ(みんなのタイピング力をあげよう)
- ・エンジニア技術記事アプリ(アーシャルのみなさんが役に立つと思ったキータなどの記事を好きに投稿できるアプリ)
- ・社内の日記帳 & Todoアプリ(Instagramみたいな日記帳)

要件定義書(作成背景)

対象者
アーシャルデザインの社員 (特にAthleteAgent-TECH事業部の社員)

サービス内容
日々の出来事を日記のように書いたり、Todoリストとして使用したりと、自由に記載できるサービス 書いたものを公開するかどうか選んで、公開を選んだ場合はログインしている社員がその書いたものを見ることができる

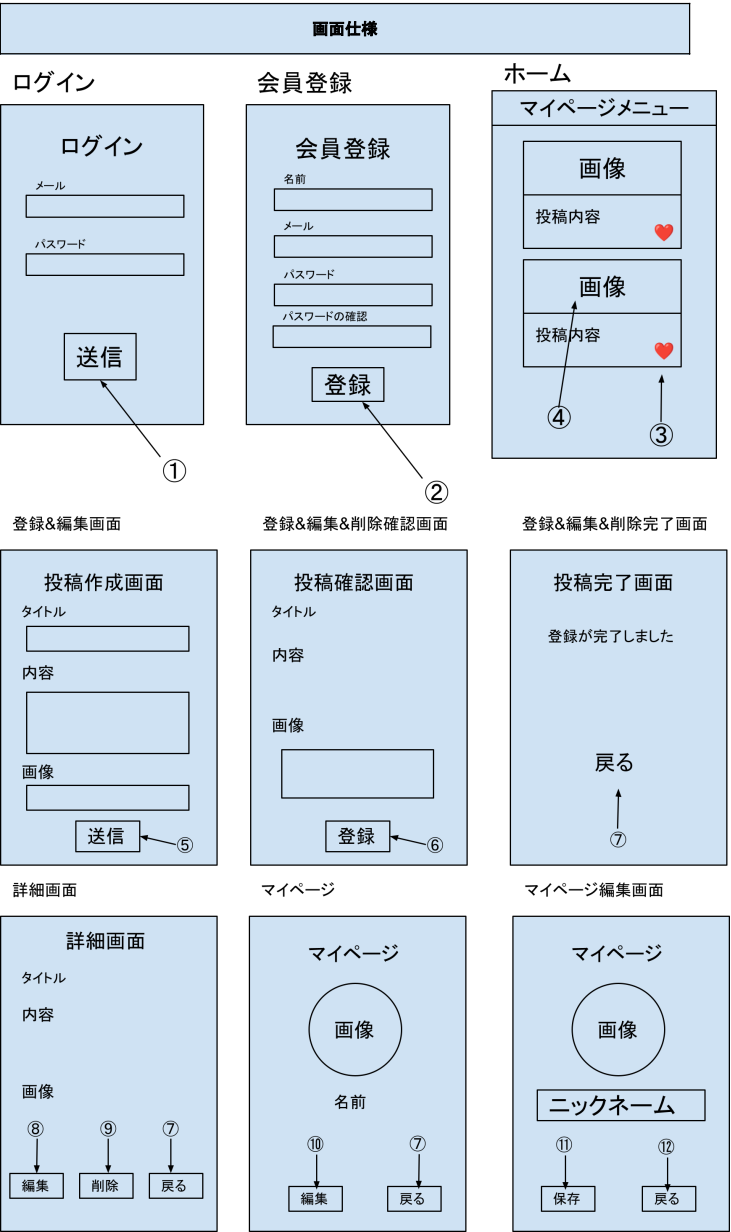
目的
交流が少ないAthleteAgent-TECH事業部の社員がこのサービスを通して、所属している社員や部署のことを知るきっかけになるといいなと考えた。

姿勢
「サービスを提供する」という目線に立ち機能を検討する。

条件
1. CRUD処理機能があること 2. 検索機能があること 3. 画像の登録,表示が行えること 4. 複数名のユーザー、権限の異なるユーザーの処理を追加すること 5. ajax または API機能を使用すること(どちらかを使う) 6. 3つ以上のDBテーブルを使用すること

サービスを完成させるためにどんな画面や機能が必要か
<ul style="list-style-type: none">・新規登録機能・ログイン機能・投稿機能<ul style="list-style-type: none">└今まで学習した内容で作成できそう・マイページ<ul style="list-style-type: none">└自分の投稿した内容が全て表示される└投稿を編集できるようにする└投稿を削除できるようにする(論理削除)・ホーム<ul style="list-style-type: none">└公開を選択したものは全て表示して、非公開は表示しない└いいね！を押せるようにする(Ajax使用)・いいね！機能<ul style="list-style-type: none">└javascriptの復習を兼ねて色々調べながら作成する

アイデアを形にするための問題点
<ul style="list-style-type: none">・Ajaxの実装方法<ul style="list-style-type: none">└どのように実装するのか？

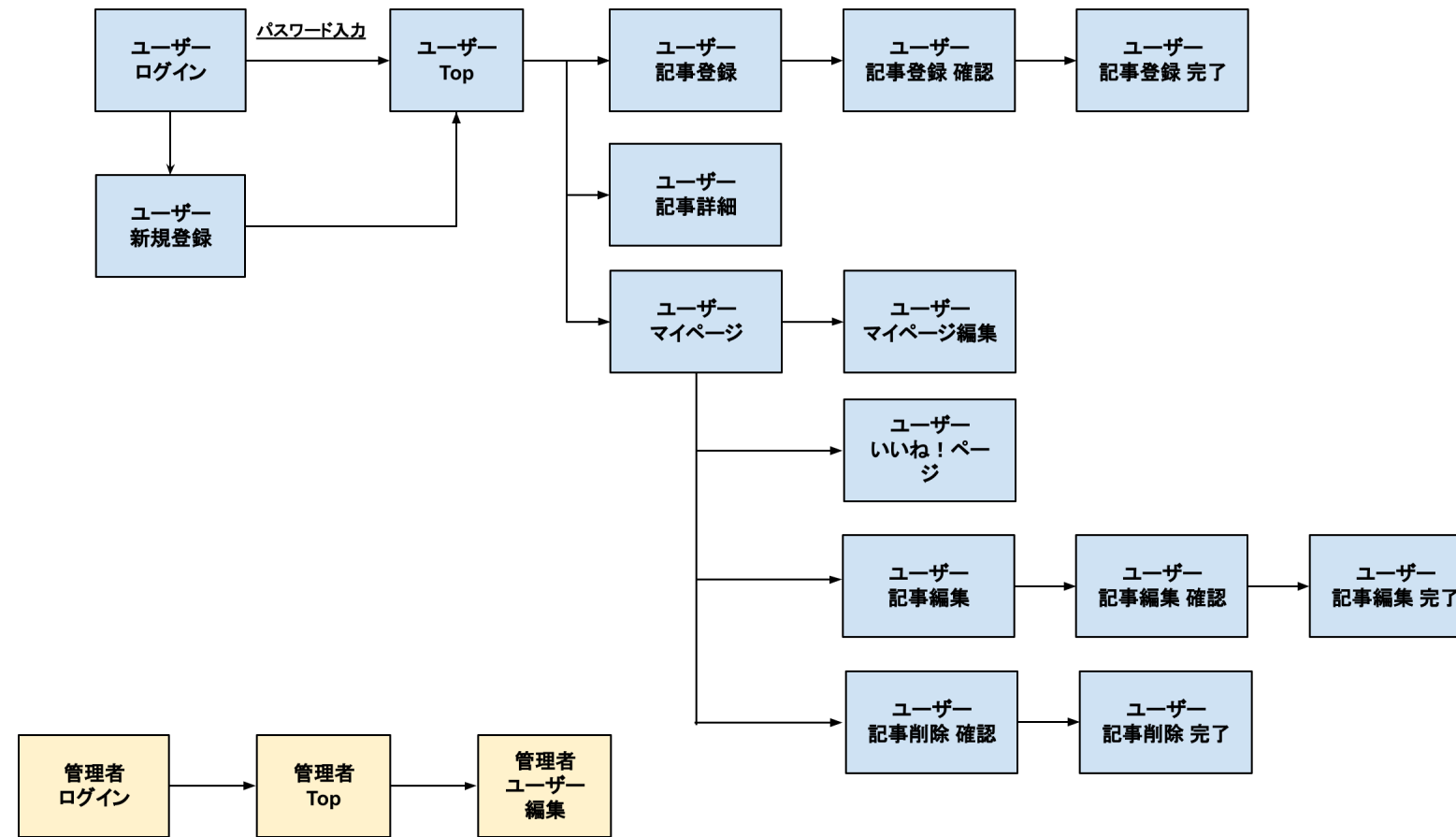


※マイページの画面に書ききれなかったが、マイページのボタン要素の下に自らの投稿が全て表示されるようにする

要素		
No.	名称	機能
1	送信	ホーム画面に遷移
2	登録	DBにユーザー情報を登録してホーム画面に遷移
3	いいね！	ボタンを押すと、いいね！IDを変更して画像を差し替える
4	画像ボタン	画像を押すと、詳細画面に遷移
5	送信	ボタンを押すと、確認画面に遷移
6	登録	DBに投稿情報を登録して完了画面に遷移
7	戻る	ボタンを押すと、ホームに遷移
8	編集ボタン	ボタンを押すと、編集画面に遷移
9	削除ボタン	ボタンを押すと、削除完了画面に遷移
10	マイページ編集	ボタンを押すと、マイページ編集画面に遷移
11	保存	ボタンを押すと、DBにユーザー情報を保存してマイページに遷移
12	マイページに戻る	ボタンを押すと、マイページに遷移

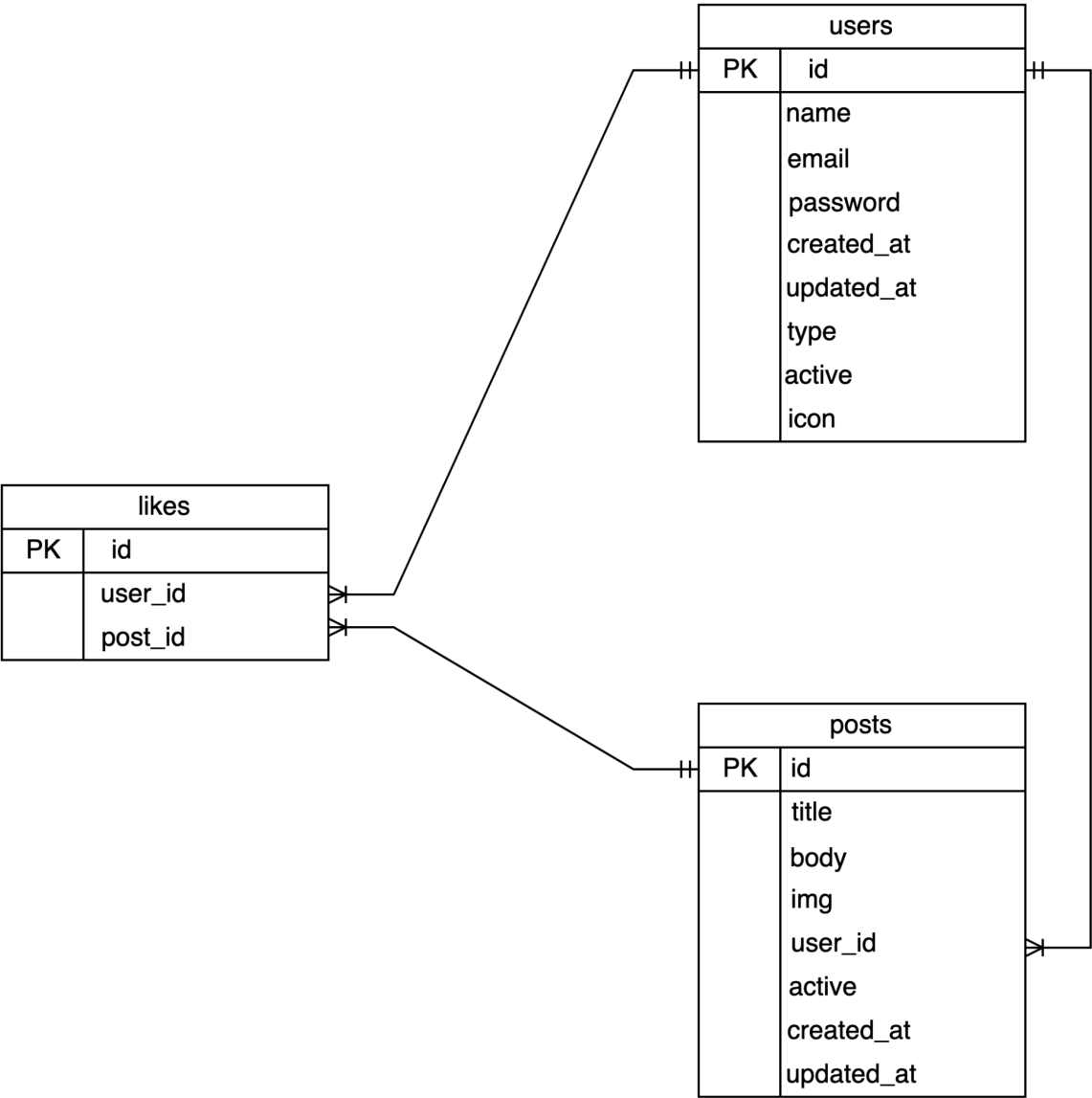
データベース仕様		
項目	内容	メモ
データベース名	diary	-
文字コード	utf8mb4	-
照合順序	utf8mb4_general_ci	-

画面遷移図



ER図

【簡易ER図】



テーブル定義書

テーブル論理名	ユーザー情報
テーブル物理名	users

項番	カラム論理名	カラム物理名	型	桁	NULL	検索項目	初期値	未編集項目	主キー	インデックス	コメント
1	登録者ID	id	int	11	NG				●	AUTO_INCREMENT	
2	登録者名	name	varchar	50	NG	●					
3	メールアドレス	email	varchar	100	NG	●				UNIQUE_1	
4	パスワード	password	varchar	100	NG						Hask::makeでパスワードをハッシュ化
5	更新日時	updated_at	datetime		NG						
6	作成日時	created_at	datetime		NG			●			
7	ユーザタイプ	type	tinyint	1	NG		1				1:一般、2:管理者
8	パスワード再発行トークン	remember_token	varchar	100	NG			●			
9	利用可能フラグ	active	tinyint	1	NG	●					1:利用可能 2:利用不可(ログイン不可)
10	アイコン	icon	varchar	256	OK						form のenctypeはマルチパートで

変更の時、未入力であれば更新なし、入力がある場合は更新

400×400のサイズ(jpg,png)

変更の時、未入力であれば更新なし、入力がある場合は更新

テーブル論理名	投稿情報
テーブル物理名	posts

項番	カラム論理名	カラム物理名	型	桁	NULL	検索項目	初期値	未編集項目	主キー	インデックス	コメント
1	投稿ID	id	int		NG				●	AUTO_INCREMENT	
2	投稿タイトル	title	varchar		NG	●					
3	投稿内容	body	varchar		NG	●					
4	画像	img	varchar		OK						
5	投稿したユーザー	user_id	int		NG						
6	いいね！のID	likes_id	int		OK						
7	更新日時	updated_at	datetime		NG			●			
8	作成日時	created_at	datetime		NG						
9	利用可能フラグ	active	tinyint		NG	●	1				

テーブル論理名	いいね！情報
テーブル物理名	likes

項番	カラム論理名	カラム物理名	型	桁	NULL	検索項目	初期値	未編集項目	主キー	インデックス	コメント
1	ID	id	int	11	NG				●	AUTO_INCREMENT	
2	いいねしたユーザー	user_id	int	11	NG				●		
3	いいねした投稿	post_id	int	11	NG				●		