

# 実験協力者募集掲示板システム 要件定義書 (Ver 3.0)

## 1. プロジェクト概要

本プロジェクトは、PythonおよびDjangoフレームワークを用いて、実験の主催者（研究者・学生等）と実験協力者（被験者）をマッチングさせるためのWeb掲示板システムを構築する。ユーザー登録・ログイン機能は実装せず、誰でも手軽に閲覧・投稿ができるオープンな掲示板形式を採用する。

## 2. システム構成・技術スタック

カテゴリ	技術要素	備考
言語	Python 3.x	
フレームワーク	Django 4.x or 5.x	フルスタックWebフレームワーク
データベース	SQLite (開発用) / PostgreSQL (推奨)	
認証方式	簡易パスワード認証	各記事ごとのパスワードによる編集・削除制限

## 3. 機能要件

### 3.1 ユーザー管理について

- ユーザー登録・ログイン機能: なし（廃止）。
- 管理者: 開発者のみがDjango管理画面（Admin）に入り、全データの管理が可能。
- 一般ユーザー（投稿者・閲覧者）: アカウントを持たず、サイトを利用する。

### 3.2 掲示板（一覧）機能

- 記事一覧表示:
  - 募集中の実験タイトル、主催者名、日時、場所などの要約を表示。
  - ページネーション機能。
- 新規投稿ボタン:
  - 誰でもアクセス可能な状態で配置。クリックで投稿フォームへ遷移。
- ソート機能:
  - 投稿日時順（新しい順/古い順）
  - 実験実施日順（近い順）
  - ※謝礼金額によるソートは不要。
- フィルタリング（検索）機能:
  - キーワード検索
  - カテゴリ
  - ステータス（募集中、締切済）

- 実施場所

### 3.3 投稿詳細・応募機能

- 詳細表示:**
  - 実験の詳細情報および、Google Formsへのリンクボタンを表示。
- 応募アクション:**
  - 「応募する（Google Formsへ）」ボタンをクリックし、外部のGoogleフォームへ遷移する。

### 3.4 投稿・編集・削除機能（パスワード認証）

- 新規投稿:**
  - 誰でも投稿可能。
  - 削除/編集用パスワード設定:** 投稿時に、投稿者が任意の**4桁の数字**を設定する。
- 編集機能:**
  - 詳細画面等の「編集」ボタンから遷移。
  - 投稿時に設定したパスワードを入力し、一致した場合のみ編集画面へ進める。
- 削除機能:**
  - 詳細画面等の「削除」ボタンから遷移。
  - パスワードを入力し、一致した場合のみ削除（物理削除または論理削除）が実行される。

## 4. データモデル・入力フィールド

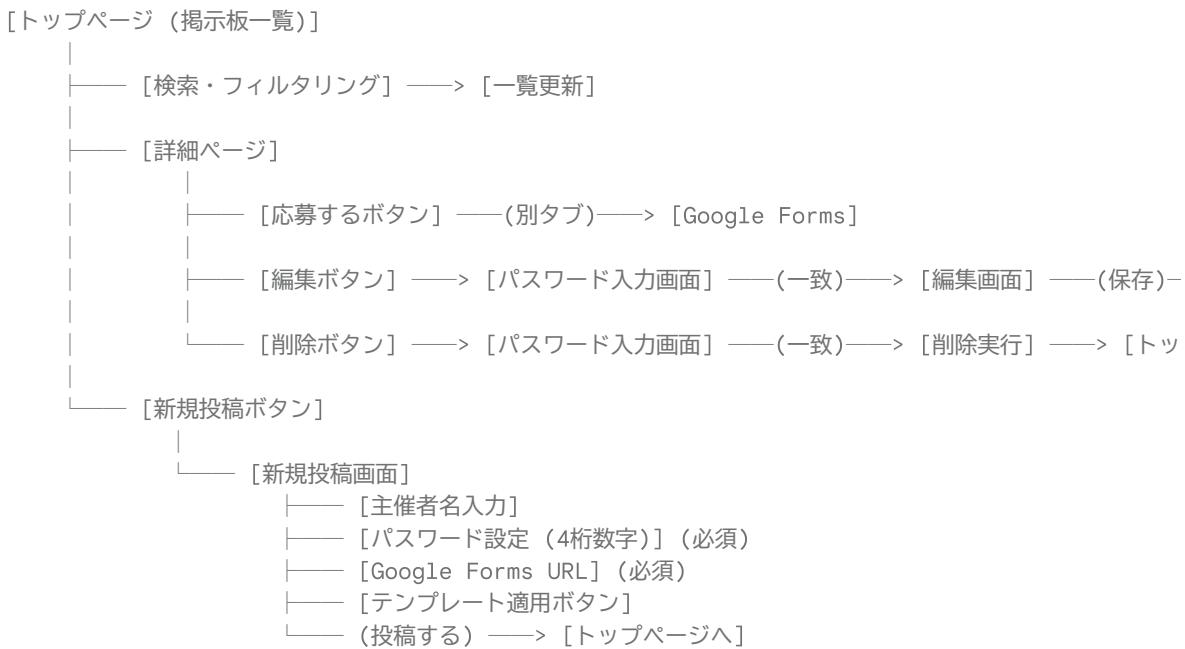
ユーザーモデルとの紐づけ（`author`）を廃止し、代わりに「投稿者名」と「パスワード」フィールドを追加します。

**モデル名:** ExperimentPost

項目名(論理名)	項目名(物理名)	データ型	必須	概要・入力例
タイトル	title	CharField	<input checked="" type="radio"/>	実験名
主催者名	organizer_name	CharField	<input checked="" type="radio"/>	追加: 誰の実験か識別するための名前（研究室名や氏名）
編集用パスワード	edit_password	CharField	<input checked="" type="radio"/>	追加: 4桁の数字(例: 1234)。RegexValidator等で数字4桁に制限。
実験概要	description	TextField	<input checked="" type="radio"/>	詳細内容
応募用URL	application_url	URLField	<input checked="" type="radio"/>	Google Forms等のURL
実施日時	schedule	DateTimeField	<input checked="" type="radio"/>	実験開始日時
所要時間	duration	CharField	<input checked="" type="radio"/>	例: 60分程度

実施場所	location	CharField	○	例: ○○キャンパス 3号館
謝礼	reward	CharField	○	表示用 (例: Amazonギフト券1000円)
応募条件	requirements	TextField	○	対象属性
募集人数	capacity	IntegerField	△	例: 10名
ステータス	status	ChoiceField	○	募集中 / 終了
作成日時	created_at	DateTimeField	○	自動設定
更新日時	updated_at	DateTimeField	○	自動設定

## 5. 画面遷移図 (UIフロー)



## 6. セキュリティ・実装上の注意点

- **パスワード管理:**

- 簡易的な掲示板であるため、パスワードは平文保存でも機能するが、可能であればDjangoのハッシュ化機能を用いて保存することが望ましい。
- 今回は「4桁数字」という要件のため、総当たり攻撃に弱いが、リスク許容範囲内（個人情報が含まれない掲示板）として実装を進める。

- **CSRF対策:**

- ログイン機能がなくても、フォーム投稿時には必ずDjangoのCSRFトークンを含めること。

- **入力検証:**

- パスワード欄が半角数字4桁であることをバリデーションする。