

†

あらまし  
キーワード

†

## Abstract Key words

### 1. はじめに

### 2. 準備

### 3. 提案手法

#### 3.1 システム概要

#### 3.2 ユースケース

#### 3.3 システムアーキテクチャ

#### 3.4 ドメインモデリング

ShelterNavi におけるドメインモデル図 1 にを示す。ShelterNavi には、図 1 で表されている 5 つのドメインが存在し、それぞれ Shelter (避難所)、User (住民・ユーザ)、Check-In (避難所へのチェックインまたはチェックアウト)、ShelterState (避難所の状態)、UserState (ユーザの状態) となっている。Check-In ドメインを通じてユーザとそのユーザが利用した避難所を関連付けさせるために、Check-In ドメインは必ず 1 つの User ドメインと 1 つの Shelter ドメインと結びつく。また、ShelterState ドメインは各避難所の状態を表しており、UserState ドメインはユーザの避難状態を示している。以下では、ShelterNavi の各ドメインを構成する要素について説明する。Shelter ドメインにおいては、システムがどの避難所か特定するための ID が必要であり、避難するユーザに向けて避難所の名称、位置情報を示す必要がある。そして、避難所の開放情報も取り扱う。以上のことから Shelter ドメインでは以下のスキーマを規定する。

- **sid**: 避難所 ID
- **name**: 避難所の名称
- **address**: 避難所の住所
- **lat**: 避難所の緯度
- **lng**: 避難所の経度
- **capacity**: 避難所の収容可能人数
- **isActive**: 避難所が利用可能か

続いて User ドメインにおいては、ユーザ情報を一意に取り扱うためのユーザ ID が必要である。また、本サービスに登録するための email アドレスとパスワード、(名前、住所: 必要性

をどう示すか)、そして混雑度を計算するための世帯人数を取り扱う。以上のことから User ドメインでは以下のスキーマを規定する。

- **uid**: ユーザ ID
- **email**: メールアドレス
- **password**: パスワード
- **name**: ユーザの名前
- **address**: ユーザの住所
- **of families**: 世帯人数

Check-In ドメインにおいては、どのユーザがどの避難所を利用しているかを管理するためにユーザ ID と避難所 ID が必要である。また、複数の避難所への多重チェックインがないかを確認するために、チェックイン時刻と、チェックアウト時刻も取り扱う。以上のことから Check-In ドメインでは以下のスキーマを規定する。

- **cid**: チェックイン ID
- **uid**: ユーザ ID
- **sid**: 避難所 ID
- **checkin-datetime**: チェックイン時刻
- **checkout-datetime**: チェックアウト時刻

ShelterState ドメインにおいては、コロナ禍における避難所での密を考慮すべく、避難所の混雑度も取り扱う。この混雑度は、避難所の収容可能人数に対しての現在の収容人数の割合から求められるので収容人数も管理する。そしてチェックイン、チェックアウトによる収容人数の変化を見るために、避難所の状態の最終更新日時も取り扱う。以上のことから ShelterState ドメインでは以下のスキーマを規定する。

- **id**: 避難所の状態 ID
- **sid**: 避難所 ID
- **num of accomodated**: 収容人数
- **congestion**: 混雑度
- **changedAt**: 更新日時

UserState ドメインでは、ユーザの避難状態の履歴を管理する。特定のユーザの履歴を見るためにユーザ ID が必要である。そして、準備中、避難済み等の避難状態を表すスキーマを取

り扱い、避難状態が変化した際の日時を記録する。なお、この UserState ドメインでは過去のデータを参照できるようにしておくために、1つのデータを更新していくのではなく、新規データを追加していく形をとる。

- **id** : ユーザの状態 ID
- **uid** : ユーザ ID
- **state** : 避難状態
- **changedAt** : 更新日時

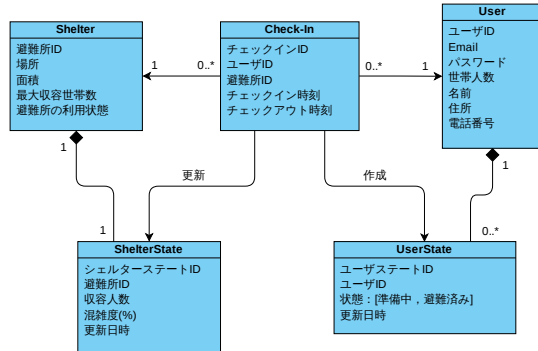


図1 ドメインモデル図

### 3.5 主要なサービス

ShelterNavi では他のアプリケーションとの連携や拡張性を考慮し、HTTP を介して外部から利用できる API を配備した。以下に API の詳細を示す。

#### 3.5.1 シェルターサービス

- **createShelter( shelterForm )** : 避難所 ID, 避難所名, 位置情報を基に避難所データを作成し、取得する。
- **getShelter( sid )** : 避難所 ID を指定することで該当する避難所データを取得する。
- **deleteShelter( sid )** : 避難所 ID を指定することで該当する避難所データを削除する。
- **clearAllShelters()** : 全ての避難所データを削除する。
- **getAllShelters()** : 全ての避難所データを取得する。
- **searchSheltersByDistance( lng, lat, distance )** : 経度, 緯度, そして距離を指定することで, 指定位置座標 (lng, lat) から半径 distance[km] 以内にある避難所データ全てを取得する。当 API では, 地球上における大圏 (大円) 距離を計算する手法を使用しており, 地球の半径を 6371[km], ユーザの経度座標を lng, 緯度座標を lat, 各避難所データの経度座標を s\_lng, 緯度座標を s\_lat とし, ユーザから半径 distance[km] 以内に存在する避難所を検索するものとして以下の計算式を用いる。

$$6371 \arccos(\cos(\text{radians}(\text{lat})) \times \cos(\text{radians}(\text{s\_lat})) \times \cos(\text{radians}(\text{s\_lng}) - \text{radians}(\text{lng})) + \sin(\text{radians}(\text{lat})) \times \sin(\text{radians}(\text{s\_lat}))) \leq \text{distance}$$

- **searchSheltersByKeyword( keyword )** : 文字列を指定することで, 全避難所データの避難所名, または避難所の住所に部分一致するものがないか検索し, 該当するデータを全て取得

する。

#### 3.5.2 ユーザーサービス

- **createUser( userForm )** : メールアドレス, パスワード, 世帯人数等のユーザ情報を元に新規アカウントを作成し、取得する。
- **getUser( uid )** : ユーザ ID を指定することで該当するユーザアカウントを取得する。
- **deleteUser( uid )** : ユーザ ID を指定することで該当するユーザアカウントを削除する。

#### 3.5.3 チェックインサービス

- **checkIn( uid, sid )** : ユーザ ID と避難所 ID を指定することで, チェックインデータを作成し, 更新日時を記録する。
- **checkOut( uid, sid )** : ユーザ ID と避難所 ID から一意に定まるチェックインデータを取得し, チェックアウト時刻を更新する。

#### 3.5.4 シェルターステートサービス

- **checkIn( uid, sid )** : ユーザ ID と避難所 ID を指定することで, 避難所の収容人数をユーザの世帯人数分増加させ, 避難所の混雑度を計算し, 更新する。また, 更新日時を記録する。
- **checkOut( uid, sid )** : ユーザ ID と避難所 ID を指定することで, 避難所の収容人数をユーザの世帯人数分減少させ, 避難所の混雑度を計算し, 更新する。また, 更新日時を記録する。

#### 3.5.5 ユーザステートサービス

- **checkIn( uid )** : ユーザ ID を指定することで, 特定のユーザの避難状態を「避難済み」にした UserState のデータを新規で作成する。また, 更新日時も併せて記録する。
- **checkOut( uid )** : ユーザ ID を指定することで, 特定のユーザの避難状態を「準備中」にした UserState のデータを新規で作成する。また, 更新日時も併せて記録する。

## 4. 実装

### 4.1 ShelterNavi プロトタイプの実装

今回は以下の開発環境で「ユーザーサービス」, 「シェルターサービス」, また「チェックインサービス」の一部の開発を行った。

- サーバ開発言語 : Java
- クライアント開発言語 : HTML5, JavaScript
- CSS ライブラリ : Bootstrap
- データベース : MySQL 8.0.20
- Web サーバ : Apache Tomcat
- Web サービスフレームワーク : SpringBoot (Java)

以下では ShelterNavi の「サインアップ」, 「ログイン」, 「避難所検索」, 「チェックイン」の各画面構成に併せて実装の詳細を述べる。

### 4.2 サインアップ

サインアップ画面を図2に示す。サインアップ画面は, ログイン時に必要なメールアドレスとパスワードからなる「ログイン情報」とログイン後, システム内で扱われる名前, 住所, 世帯人数, 電話番号 (任意) からなる「個人情報」の2つの要素で構成されている。これらの情報を入力したうえで図2下部の登録ボタンをクリックすれば, ユーザ新規登録の完了となる。

4.3 ログイン

4.4 避難所検索

4.5 チェックイン



Shelter Navi

ユーザ新規登録

下記の情報を入力して登録ボタンを押してください

ログイン情報

メールアドレス

you@example.com

パスワード

password

パスワード (確認)

password

個人情報

姓

名

住所

神戸市○○区○○町1-2-3-456

世帯人数

1

電話番号 (緊急時連絡先)

090-999-9999

☐ プライバシーポリシーに同意して登録する

登録

© 2020 H. Nakada, T. Murotani, Nakamura Lab., Kobe University

図 2 新規登録画面



Shelter Navi

With/Afterコロナ時代の避難所ナビゲーション

ログインしてください

メールアドレス

パスワード

☐ 情報を記憶する

ログイン

または

新規登録

アプリ説明

© 2020 H. Nakada, T. Murotani, Nakamura Lab., Kobe University

図 3 ログイン画面

Shelter Navi 避難所検索 チェックイン マイ避難所 中田 大翔

避難所検索

現在地に移動

地図 航空写真



#	避難所名	住所	混雑状況
1	神戸大学工学部	神戸市灘区六甲台町1-1	26
2	神戸大学農学部	神戸市灘区六甲台町1-1	26
3	高羽小学校	神戸市灘区高羽町3-11-11	39
4	親和女子高等学校・親和中学校	神戸市灘区土山町6-1	65
5	神戸松蔭女子学院大学	神戸市灘区篠原泊母野山町1-2-1	26
6	鶴甲小学校	神戸市灘区鶴甲2-10-1	39
7	鷹匠中学校	神戸市灘区高徳町2-2-19	65
8	六甲小学校	神戸市灘区八幡町4-4-1	39
9	神戸大学発達科学部	神戸市灘区鶴甲3-11	26

避難所IDで検索する

検索

© 2020 H. Nakada, T. Murotani, Nakamura Lab., Kobe University

図 4 避難所検索画面

## 5. 考察・評価

## 6. おわりに

### 文 献

[1]