社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

コロナ時代の密を考慮した避難所ナビゲーションアプリの開発

中田　大翔†　室谷　敏生††　中村　匡秀††

† 兵庫県立姫路西高等学校
†† 神戸大学大学院システム情報学研究科
E-mail: †{hiroto.nakata,musan}@ws.cs.kobe-u.ac.jp, ††masa-n@cs.kobe-u.ac.jp

あらまし　日本は災害大国であり, 毎年甚大な被害が出ている. 2020 年には COVID-19 の世界的流行により, 災害時の避難所において住民の受け入れ人数の削減などが行われ, 避難所に収容できないケースが発生している. 本研究では, 住民の自助によって避難所での密の形成を回避するアプリケーション Shelter Navi を提案する. Shelter Navi は自治体内に存在する避難所の場所と混雑度合いを管理し，地図上に可視化して，災害時に住民が密を避けて分散避難するための情報を提供する．住民が避難所に「チェックイン」すると，Shelter Navi は混雑度合いをリアルタイムに更新する．これによって，各避難所に特別な設備を必要とすることなく，避難所での密を考慮した避難が可能となる．本稿では，Shelter Navi のユースケースを定義し，ドメインモデルおよびサービスの設計を行う．さらに，Shelter Navi のプロトタイプを Spring Boot および Bootstrap を活用したモバイル Web アプリケーションとして実装する．

キーワード　災害，Web アプリ，新型コロナウイルス，避難意識, 避難所，自助

Development of Shelter Navigation with Considering Three Cs in the Corona Age

Hiroto NAKADA†, Toshiki MUROTANI††, and Masahide NAKAMURA††

† Himeji West High School
†† Graduate School of System Informatics, Kobe University
E-mail: †{hiroto.nakata,musan}@ws.cs.kobe-u.ac.jp, ††masa-n@cs.kobe-u.ac.jp

Abstract　Japan is a disaster-prone country, and tremendous damage occurs every year. Due to the global pandemic of COVID-19 in 2020, evacuation shelters have been forced to reduce the number of residents, in order to reduce the risk of infection. Some evacuation shelters have been unable to accommodate evacuators due to full capacity. In this study, we propose a novel application, Shelter Navi. It aims to avoid the formation of dense crowd in evacuation shelters by self-efforts of citizens themselves. Shelter Navi manages the location and the congestion status of every shelter within a local government, and visualizes the information on a map. When a disaster occurs, citizens check the information with mobile phones, and evacuate to vacant shelters. As a citizen “checks in” to a shelter, Shelter Navi updates the status at real time. Thus, the app allows crowd-aware evacuation without any special equipment in the shelter. In this paper, we first define use cases of Shelter Navi, and then conduct domain modeling and service API design. Finally, we implement a prototype of Shelter Navi as a Web application, using latest application frameworks Spring Boot and Bootstrap.

Key words　disaster, Web application, COVID-19, preparedness, evacuation shelters, self-effort

1. はじめに

日本は災害大国であり，毎年大規模な自然災害が発生している. 近年では，いわゆる異常気象による豪雨により甚大な被害が発生しており，命を落とす人も後を絶たない. 2018 年 7 月の西日本で発生した豪雨災害では，死者 200 人を超える被害が出た [1]．犠牲者が出た地域の多くは，洪水浸水想定区域や土砂災害警戒区域内であり，避難行動を促す情報が事前に発令されていた. つまり，特に豪雨災害においては，住民が逃げ遅れによって死亡している事例が多い. 逃げ遅れの主な要因は，「自分は大丈夫だ」と思い込む正常性バイアス等の心理効果による避難意識の欠如 [2]〜[4]，災害時における適切な行動が分からないなどの知識不足，避難場所の確認不足などが挙げられる [5]. さらに 2020 年に起こった新型コロナウイルス感染症の流行に

よって，感染対策を考慮した新たな避難所運営が必要になっている [7]．その一環として，三密回避のための避難所の受け入れ人数制限・管理があり，避難してきた住民が避難所に入れなくなる事態が想定される．実際に 2020 年 9 月九州に台風 10 号が直撃し避難所が開設された際に，避難してきた住民が 2 か所続けて受入拒否された事例が発生している [8]. このような問題に対して，独自に対策を行なっている自治体もある．宮崎県日南市では，飲食店などの混雑状況を配信するアプリケーション VACAN を用いて, 避難所の混雑状況の配信を行い，一定の効果を上げている [9]．しかしながら，避難所の混雑状況は，自治体職員の手作業によって計測・更新されている．したがって，コロナ時代に必要な新たな施策も，各自治体に任せきりになっており，職員の職務負担の増大につながっているのが現状である．このような背景の下，我々は「コロナ時代に災害が起きた場合，住民が自治体に頼ることなく，自分たちで適切な避難所へ分散避難できないか？」をリサーチクエスチョンに設定して研究を進めている．本研究では，住民の自助によって避難所での密の形成を回避し，迅速且つ安全な避難の実現を支援するモバイル・アプリケーション Shelter Navi を提案する．Shelter Navi は，クラウドサーバで自治体内の避難所の場所と混雑状態を管理し，地図上に可視化して，災害時に住民が密を避けて分散避難するための情報を提供する．住民が避難所に「チェックイン」すると，Shelter Navi は混雑度合いをリアルタイムに更新する．これによって，各避難所に特別な設備を必要とすることなく，避難所での密を考慮した避難が可能となる．本稿では，Shelter Navi のユースケースを定義し，ドメインモデルおよびサービスの設計を行う．さらに，Shelter Navi のプロトタイプを Spring Boot [10] および Bootstrap を活用したモバイル Web アプリケーションとして実装する．Shelter Navi によって，住民の避難意識の向上と，With/After コロナ時代の避難所運営の効率化につながることが期待できる．

2. 準　　備

2.1 日本における災害避難

近年わが国では，異常気象などの影響により「数十年に一度」と言われるような大規模な豪雨災害が頻繁に発生しており，毎年多くの犠牲者が発生している． 2018 年 7 月の西日本豪雨災害では，避難勧告が発令されていたにもかかわらず，人的被害が多く発生したことが報告されている．[1] 犠牲者の中には「逃げ遅れ」によるものが多く存在した．災害時における住民の避難を促すための情報として，気象庁から発令される大雨・暴風・洪水等の気象警報，各自治体から発令される避難勧告並びに避難指示等がある．避難指示に関しては，強制力はないため，避難するかどうかは，避難することによる危険などを踏まえた住民の判断に任されている．しかし，このような自身の身に危険が迫っている場合は，大量の情報の処理を時間的制約がある中で正確に行うことを迫られることなどによる強いストレスにより，冷静さを保つ目的で，平常時と同じリスク評価，つまり事態を楽観視してしまう傾向である「平常性バイアス」が働く．これにより，自身に都合の良いように解釈をしてしまい，「自分は大丈夫だ」といったような思考が生まれ，結果として意思決定や避難行動の遅れにつながっていると考えられている．[2] 加えて，避難する避難所の確認ができていない，避難経路がわからないなどの住民の事前準備や意識の低さも避難率の低下につながっている．

2.2 コロナ時代の避難所運営

2020 年，新型コロナウイルス感染症の世界的流行により，社会のあらゆる場面で様式の変化が求められた．防災の面では，避難所での感染対策が急務で進められ，各自治体から感染対策が盛り込まれた新たな避難所運営マニュアルが発行された．具体的には，避難スペースの設置レイアウト例（収容人数，間隔など）や避難所受け入れ前の検温・問診などの実施などが新たに示された．また，密集を避けるため，避難所当たりの収容人数を大きく制限せざるをえなくなった．このような感染対策により，状況がわからずに避難してきた住民が避難所に入れてもらえないケースが懸念されている．2020 年 9 月九州に台風 10 号が直撃し避難所が開設された際に，避難してきた住民が 2 か所続けて受入拒否された事例が発生している [8]．このような問題に対し，独自に対策を行なっている自治体もある．宮前県日南市では，飲食店の混雑状況などを配信するアプリケーション「VACAN」を用いて，避難所の混雑状況の配信を行っている．しかしながら，避難所の混雑状況は，自治体職員の手作業によって計測・更新されている．このように，コロナ時代における避難所運営の新たな施策も試行錯誤的に行われてきているが，基本的には自治体に任せきりになっており，職員の職務負担の増大につながっているのが現状である．さらには，このような自治体の対策に住民が依存してしまい，住民の自助が抑制されてしまうことも懸念される．

2.3 リサーチクエスチョン

以上を踏まえて，我々は以下のリサーチクエスチョンを設定し，それに答える手法を研究している．

- **RQ**「コロナ時代に災害が起きた場合，住民が自治体に頼ることなく，自分たちで適切な避難所へ分散避難できないか？」

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．Shelter-Navi 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

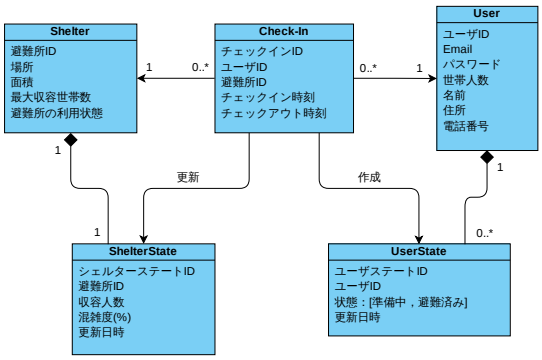

図 1　ドメインモデル図

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．

$$6371 \arccos(\cos(radians(lat)) \times \cos(radians(s\_lat)) \times \cos(radians(s\_lng) - radians(lng)) + \sin(radians(lat)) \times \sin(radians(s\_lat))) \leqq distance$$

- **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

ログイン画面を図 3 に示す．ログイン画面では 4.2 で述べた「ログイン情報」であるメールアドレスとパスワードを画面中央にある項目に入力し，ログインボタンをクリックする．ユーザとして登録済みの正しいメールアドレスとパスワードが入力できていれば，ログイン後に避難所検索画面に移行する．ログインが成功した際は，ユーザごとに権限が与えられる仕組みになっており，通常付与される権限では 3.5 で紹介した API の一部しか扱えないため，自身の登録した情報が他ユーザから閲覧，更新，削除されることはない．

4.4 避難所検索

避難所検索画面を図 4 に示す．避難所検索画面では画面上部に青いマーカーを現在地とした地図が表示されており，現在地から半径 1[km] 以内の避難所を赤いマーカーで配置している．避難所の表示は 3.5 で紹介した searchSheltersByDistance を利用してリアルタイムでの更新を行っている．避難所のマーカーは．クリックすることで避難所の詳細を表示することもできる．また画面下部のリストは，地図上の避難所を現在地からの距離が近い順に表示したものである．リストの各項目は左から地図上のマーカーの番号，避難所名，避難所の住所，混雑状況の 4 つである．最後に，画面最下部に検索バーでは避難所の ID，もしくは避難所名ないし，避難所の住所に含まれる文字を入力することで避難所の検索をすることができる．

4.5 チェックイン

チェックイン画面を図 5 に示す．チェックイン画面では，チェックインする避難所を「現在地」,「エリア」,「地図」のいずれかから選択することができる．「現在地」から選択する場合，画面上部の青色の「現在地を取得」ボタンをクリックすれば，現在地に存在する避難所を選択することができる．「エリア」から選択する場合は，画面中央の「エリアから選択」の項目である都道府県，市町村，地域をプルダウンメニューから選択することで避難所のプルダウンメニューに条件に該当する避難所リストが現れる．「地図」から選択する場合は，画面下部の地図上にある赤いマーカーをクリックすることで避難所の選択が可能になる．これらのいずれかの方法で避難所を選択してから画面右下の「チェックインする」ボタンを押すことで避難所へのチェックインが完了する．

4.6 チェックアウト

チェックアウト画面を図 6 に示す．いずれかの避難所にチェックイン済みの場合，チェックアウト画面左下には「現在の避難所：「※チェックイン済みの避難所名」からチェックアウトしますか？」の 1 文が表示される．画面右下の「チェックアウトする」ボタンを押すことで，現在チェックインしている避難所からのチェックアウトが完了する．

5. 考察・評価

6. おわりに

文　　献

[1] 国土交通省 “平成３０年７月豪雨災害の概要と被害の特徴”

図 2　新規登録画面

図 3　ログイン画面

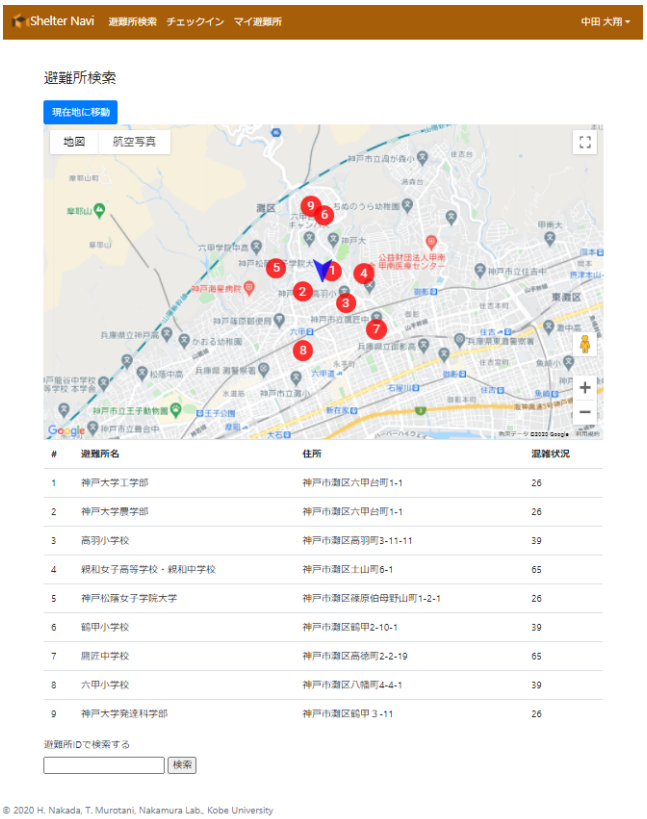

図 4　避難所検索画面

図 5　チェックイン画面

[2] 菊池聡,“非常時の思い違いと批判的思考” 日本科学教育学会年会論文集, Vol.35, pp.9-10, 2011.
[3] 国土交通省 “住民自らの行動に結びつく災害情報の提供へ ～危機感が伝わる、メディアとの連携策をとりまとめ～” 2018.12.11
[4] 皆川勝、中村遼太、高橋翔天 “極低頻度の災害に対する避難行動の社会心理学的な考察” 土木学会論文集 F 6 Vol.72 No.2 I191 I198 2015.7.10
[5] 田中重好“東日本大震災を踏まえた防災パラダイム転換”Vol.64 No.3 p.342-365 2013.
[6] 齋藤, 美絵子 “災害リスクコミュニケーションのためデジタルツールの効果に関する研究” 2018.03.23
[7] 内閣府 “新型コロナウイルス感染症を踏まえた災害対応のポイント【第１版】について”
[8] 毎日新聞 “避難所「先着順で満員、2 回も振られる」台風 10 号、コロナ対策で収容人数の減少で” 2020.09.07
[9] 株式会社バカン “株式会社バカンと宮崎県日南市、災害発生時に IoT を活用して避難所の混雑情報配信を支援する協定を締結” 2020.08.03
[10] Spring “Why Spring?” https://spring.io/why-spring

図 6　チェックアウト画面