

Web Bundles を用いた災害時の申請 システムの検討

龍谷大学 理工学部 情報メディア学科

T170494 高山 順暉

指導教員 三好 力 教授

内容概要

大災害が発生した際、被災者は支援金の受け取りや、公共料金の支払い免除などの支援を受けることができる。しかし、災害の混乱のなか、役所で申請を行うとさらに不必要な混乱を招く可能性がある。そこで、オンラインで申請することが提案されているが、大災害の際にはインターネットに接続できない場合がある。本研究ではそのようなオフライン環境で動作する、行政サービスの電子申請システムの開発を行った。オフラインでの電子申請を可能にするために web bundles と JavaScript を利用した。また、非常時の申請を想定し、スマートフォンなどの携帯端末からの申請を可能とした。

目次

第1章 はじめに	1
1.1 災害発生後、避難する際に持っていくもの	1
1.2 スマートフォン普及率	2
1.3 問題点	2
第2章 既存技術及び類似システム	3
2.1 Web bundles	3
2.2 Bluetooth	3
2.3 公開鍵暗号方式.....	4
2.4 DTN	5
2.5 e-Gov	5
2.6 PWA	6
第3章 提案手法	8
3.1 概要.....	8
3.2 JavaScript による HTML ファイル生成.....	9
第4章 実験.....	10
4.1 実験目的.....	10
4.2 実験環境.....	10
4.3 システム構築.....	10
4.3.1 バンドルファイル(wbn ファイル)	10
4.3.2 申請ファイル(HTML ファイル).....	11
4.3.3 自己証明書の作成	12
4.3.4 自己証明書の付与した html ファイルから har ファイル作成	14
4.3.5 bluetooth でファイルの転送	14
4.3.6 データベースへ格納	15
4.4 考察.....	16
第5章 結論.....	17

謝辭	18
参考文献	19
付録	20

第1章 はじめに

日本では、大雨や台風、地震などの自然災害が多く発生している。その中でも特に大規模な災害が起こった際、重要なインフラストラクチャーが機能しなくなる可能性が非常に高い。実際、熊本地震や東日本大震災などの自然災害発生時にインターネットが使用することができない事例が多く発生している。災害発生後、被災者の方々の多くは避難所に避難しており、その際に役所に提出する書類は、一人一人手書きで記入する場合がほとんどである。スマートフォンで申請書類の記入と提出が可能になれば、効率的になるのではないかと考えられる。避難所に携帯電話(スマートフォン)を持って行く人が多いと考えられるため、書類申請にアプリケーションではなく、web ベースの入力をスマホでできるシステムが求められる。

1.1 災害発生後、避難する際に持って行くもの

20代から60代の男女1,000人に「災害発生後、避難する際に持って行くもの」についてアンケートした結果、2位の「お金・財布」(85.8%)を抑え、1位は「携帯電話(スマートフォンを含む)」(86.9%)という結果となった。続いて「水」(62.0%)、「充電器」(56.1%)、「食料」(52.2%)となった。携帯電話と充電器が上位となったことで、避難時には情報収集や連絡手段の確保などが特に重視されていることが明らかになった。[2]

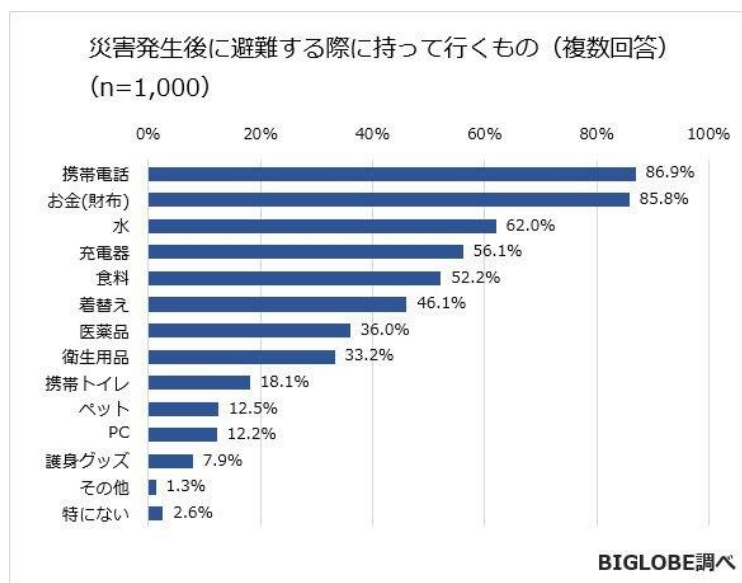


図 1.1 災害発生後に避難する際に持って行くもの

1.2 スマートフォン普及率

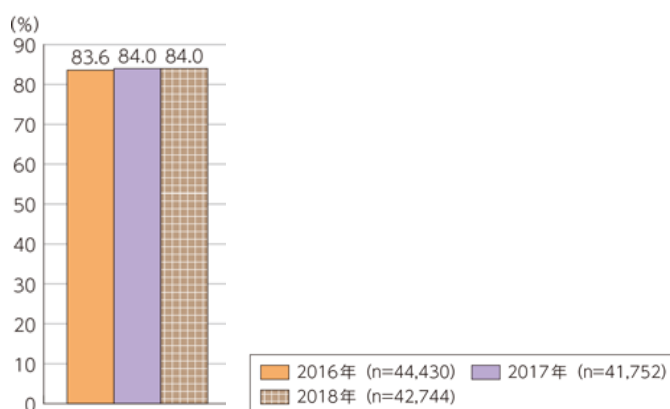


図 1.2 携帯電話普及率

2018年の時点でスマートフォンの保有率は約84%となっている。仕事をする上でスマートフォンが必要なことが多く、また学生などでも情報収集や SNS の利用でスマートフォンを持っている人が多い。

1.3 問題点

手書きで記入すると、作業スピードが落ち効率が悪くなるとともに、書き間違いや読み間違いなどの人為的なミスが生じるリスクが高い。また紙媒体で集計すると時間と手間がかかってしまう。災害発生時には迅速な対応が求められるのに、手書きによる作業をすることで必要な物資や情報が遅れる可能性が生じるため Web 形式での申請が理想である。しかし大規模災害発生時にはインフラストラクチャーが機能せず、インターネットが利用できない。専用アプリか、Web ページでの申請が理想的ではあるが、専用アプリがあっても、被災者全員にアプリが入っていないと利用できない。災害発生時にアプリをダウンロードしようとしてもインターネットが利用できないので、ダウンロードも不可能になる。同様に Web ページもインターネットが使えないとアクセスできないので利用できない。

第2章 既存技術及び類似システム

2.1 Web bundles

Web Bundle は Web ページを表示させるために必要なすべてのコードとファイルをパッケージ化した、実験的な結合ファイルである。Web Bundle は HTML、CSS、Javascript、画像、フォントなどすべての Web サイトのリソースを chrome などのブラウザが Web サイトを表示するために使用する単一の「.wbn」ファイルにパッケージ化するものである。データを積極的にキャッシュすることを可能とし、必ずしも端末がオンラインであることを要求はしない。Web Bundles は現在のところ Chrome の実験的機能として提供されている。[1]

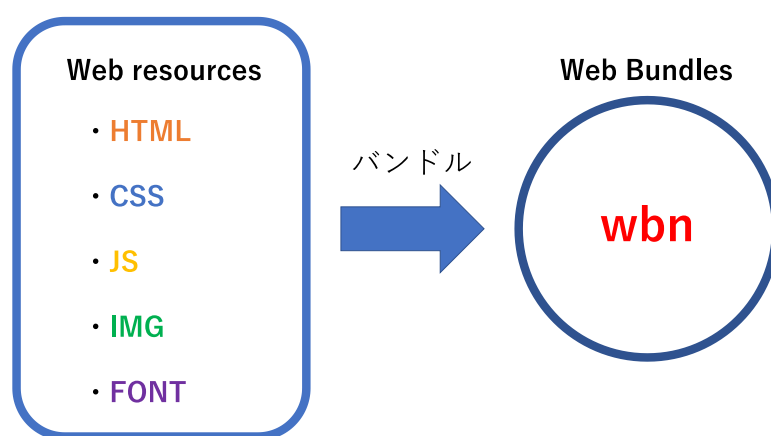


図 2.1 Web bundles の概略図

2.2 Bluetooth

Bluetooth とは、デジタルデバイスの近距離無線通信規格の一種であり 2.4GHz 帯の周波数を利用してデジタルデバイス同士の通信を行うもののことである。Bluetooth は国際標準規格となっているため、多くのデバイスが対応しており、対応しているデバイスであれば国やメーカーを問わず接続することが可能である。

2.2.1 プロファイル

Bluetooth はオーディオデバイスやキーボードなど、様々なデバイスでの通信に利用されるため、接続する機器の種類によって通信時の動作や送受信されるデータの内容を規定する必要があり、それをプロファイルと呼ぶ。以下はその中で代表的なプロファイルである。

- Hands-Free Profile(HFP)

HFP はヘッドセットなどのオーディオ機器でハンズフリー通話を行うためのプロファイル

である。音声の受信とマイクの入力を実現する Headset Profile の機能に加えて、通話の発信と着信応答の機能を有する。

- Human Interface Device Profile(HID)

HID はマウスやキーボードなどのヒューマンインタフェースを無線接続するためのプロファイルである。このプロファイルによってスマートフォンなどの端末でもキーボード入力を行うことができる。

- File Transfer Profile(FTP)

FTP はデバイス同士でのファイルデータ転送を実現するためのプロファイルである。このプロファイルによって、Wi-Fi などを経由せずに手軽にファイル転送を行うことができる。

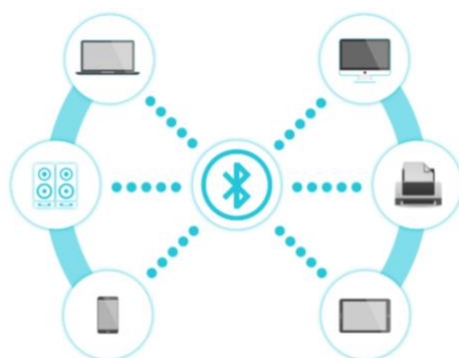


図 2.2 bluetooth イメージ図

2.3 公開鍵暗号方式

公開鍵暗号方式とはデータの暗号化、複合化を行う際にそれぞれ別個の鍵を用いる暗号化方式である。使用される 2 つの鍵はそれぞれ公開鍵と秘密鍵と呼ばれる。公開鍵は誰にでも取得可能なオープンな鍵であるのに対して、秘密鍵は送信者または受信者の片方のみが保持している鍵である。ある公開鍵で暗号化されたデータは、それと対になっている秘密鍵でのみ復号化が可能である。逆に、ある秘密鍵で暗号化されたデータは、それと対になっている公開鍵でのみ復号化が可能である。この暗号化方式には、公開鍵で暗号化し、秘密鍵で復号化する手法と、秘密鍵で暗号化し、公開鍵で復号化する手法の 2 種類があり、これらは目的によって使い分けられる。送信者が公開鍵を用いて暗号化することで通信傍受への対策となる。公開鍵で暗号化したデータの通信が傍受された場合、傍受者は秘密鍵を所持していないと復号化を行うことができないため、機密性の高いデータのやり取りが可能である。送信者が秘密鍵を用いて暗号化することによって、データ改ざんへの対策となる。送信者が秘密鍵で暗号化することで、通信の完全性を保つことが出来る。送信者が秘密鍵で暗号化したデータと暗号化前のデータを送信し、受信者が公開鍵で復号化する。この暗号化データは秘密鍵でのみ作成可能なので、複合

化したデータと暗号化前のデータを比較することで、改ざんやなりすましが行われていないことが確認できる。

マイナンバーカードを用いた行政サービス申請の電子署名では、秘密鍵で暗号化を行う方式が用いられている。具体的な手法としては、利用者側で行政サービスの申請書をハッシュ化し、そのハッシュ値をマイナンバーカード内秘密鍵で暗号化する。その後、申請書の原文と暗号化されたハッシュ値を併せて送信する。最後に行政機関側が暗号化されたハッシュ値を公開鍵で復号化し、併せて送信された申請書のハッシュ値と比較することで改ざんが行われていないことを確認する。ここで、マイナンバーカード内秘密鍵はカード外に読み出すことはできず、無理に読みだそうとすると IC チップが壊れる仕組みとなっている。また、秘密鍵を利用するためのパスワードは 3 回入力を誤るとカードがロックされてしまうため、非常に安全性が高い認証手法となっている。

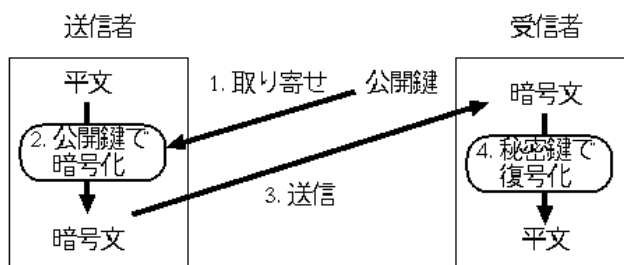


図 2.3 公開鍵暗号方式図

2.4 DTN

Delay Tolerant Networking (以下 DTN)とはエンドツーエンドでの情報伝達時において、通信の切断や大きな伝送遅延が発生する状況下でも情報伝達を実現するために提案される通信方式である。DTN の例として、蓄積型通信が挙げられる。蓄積型通信とは送信ノードから宛先ノードまでの中継転送において、中継ノードが次ノードとの通信ができない場合に通信可能となるまでデータを一時的に保持する通信形態である。このとき、中継ノードを人や車両などが移動して運搬し次ノードまで近づいて近距離通信を行うことでデータ転送を行う方式を蓄積運搬型通信という。[4]

2.5 e-Gov

e-Gov(イーガブ「電子政府の総合窓口」以下 e-Gov)とは、政府が運営する行政情報ポータルサイトを指す。e-Gov では国の行政機関に対する申請・届出等をオンラインで行える。従来書面によって行っていた申請・届出等をインターネットを利用して、オンライン申請により行政窓口に出向くことなく、自宅やオフィスからインターネットでの申請・届出や

証明書の取得が可能となる。e-Gov で申請・届出等を行う場合、マイナンバーカードが必要になる。マイナンバーカードには公的個人認証サービスの電子証明書が標準搭載されており、申請・届出の際に IC カードリーダーライターにマイナンバーカードを読み込ませることで申請・届出が可能となる。

2.6 PWA

Progressive Web Apps(以下 PWA)とはモバイル向け WBE サイトをスマートフォンアプリのように使えるようにするための仕組みのことである。PWA はそれ自体が何か特殊な一つの技術というわけではなく、レスポンシブデザイン、HTTPS 化など、Google が定める要素を備えた Web サイトであり、オフラインやプッシュ通知に対応するためのブラウザ API を利用している Web サイトを PWA と呼ばれる。[6]

2.6.1 PWA の機能

①ユーザーの手元の端末にインストール可能

モバイル端末のホーム画面にアイコンを設置できるため、ユーザーはアイコンをタッチするだけで Web サイトを閲覧することができる。App store や Googleplay などでダウンロードできるアプリとは違い PWA はストアを経由しないため、PWA を実装したサイトをリリースすればすぐにアイコン設置が可能である。また OS に依存しないため Android や iOS に個別対応する必要がない。

②キャッシュの利用

PWA ではキャッシュによってページを先んじて読み込めるため、ページの表示速度が早くなりユーザーがページを移動しやすくなる。また、インターネットに接続していなくてもキャッシュを利用してページを表示するため、オフラインでもサイトを閲覧できる。オフラインの間に入力されたデータをオンラインになったら送信するといったこともできる。

③検索結果や URL から、通常のサイトと同じようにアクセスすることができる

アプリ的な機能が取り上げられがちだが、もともとは Web サイトなので、通常の Web サイトと同じようにアクセスすることが出来る。知人とページをシェアするときはアプリとは違い URL を送ればページを共有可能であり、検索エンジンからの PWA 対応のサイトを見つけることも可能である。

PWA を実装しているサイトをタブレット端末にダウンロードを行い、ストアで提供されているアプリケーションと同じように利用することができることを確認した。しかし、オフライン環境で動作させると、ホーム画面の表示はされるが、そこから他のページに移動することができなかつたのである。[8]

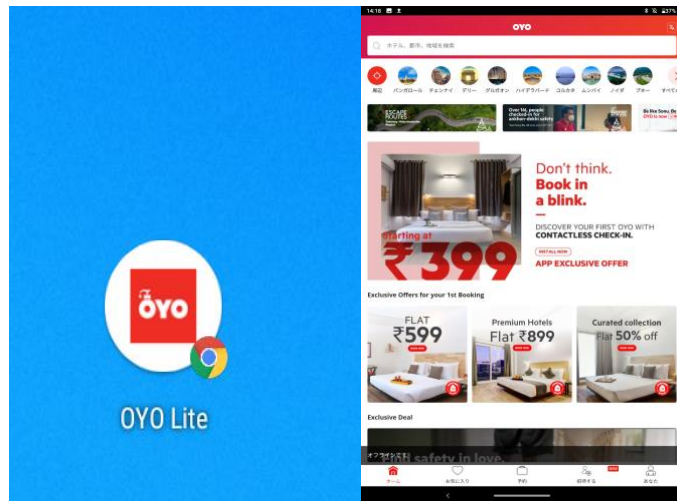


図 2.4 PWA が実装されたサイトのダウンロード

PWA を利用するために事前にオンライン環境でダウンロードすることが必要なため、災害発生時などの緊急時にインターネットが利用できなくなっている状態では、その機能を十二分に利用することができない。よってオフライン環境でもファイルの送受信や web ベースでの申請が可能なシステムの開発が必要になる。

第3章 提案手法

3.1 概要

本節ではオフライン行政サービス申請システムを提案する。以下に本システムの具体的な申請手順を示す。図 3.1 はそれを示した図である。ここでは災害時に設置される避難所での申請を想定している。なお、本システムでは災害によってインターネットへの接続が不可能となっており、避難所内でのローカルネットワークも構築されていないと想定する。

1. 行政サービス申請用のウェブページを Web Bundles を用いてバンドルする。ここで、申請用ページには JavaScript が埋め込まれており、フォームの送信ボタンを押すと、新たな HTML ファイルを生成する。この新たな HTML ファイルにもフォームがあり、はじめの申請フォームに入力した内容が初期値として格納されている。
2. 1.でバンドルされたファイル(以下バンドルファイル)を USB などの記憶装置に保存し、避難所サーバーまで運搬する。
3. Bluetooth ファイル転送を用いてバンドルファイルを申請者のデバイスへ転送する。
4. 申請者がバンドルファイルを展開し、申請用ウェブページのフォームに申請内容を入力する。
5. 申請内容入力後、送信ボタンを押すことで入力内容をフォームの初期値とした HTML ファイルを生成する。
6. Bluetooth ファイル転送を用いて 5.で生成した HTML ファイル(以下申請ファイル)を避難所サーバーに転送する。
7. 記入された HTML ファイルを USB 等の記憶装置で役所サーバーまで運搬する。
8. 役所内で記入された HTML ファイルからデータベースに格納する。

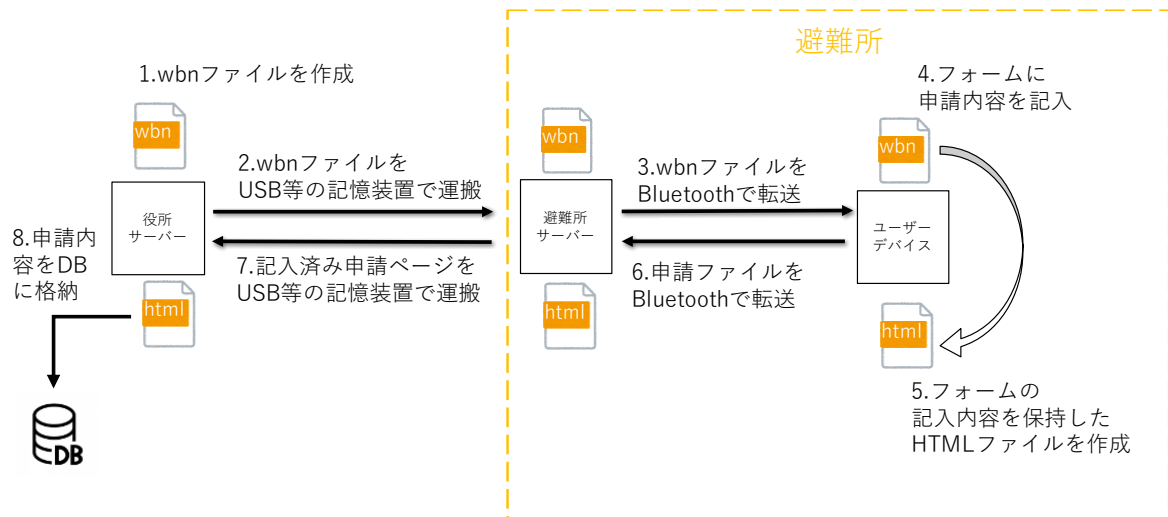


図 3.1 本研究のシステムの流れ

3.2 JavaScript による HTML ファイル生成

本研究ではオフライン状態で専用アプリを利用しない申請を可能とするために、ウェブ形式で申請ページを作成し、Web Bundle でバンドルする。しかし、避難所などでローカルネットワークの構築が行われていないと記入した申請内容を送信することができない。そこで、JavaScript の document プロパティを利用してフォームの入力内容を取得し、それらをフォームの初期値とする新たな HTML を作成する。ここで新たに生成される HTML ファイルのフォーム送信先は役所サーバーの PHP ファイルやデータベースなど任意に指定できるため、役所に持ち帰った申請ファイルを展開して送信ボタンを押すだけでデータの収集が完了する。

第4章 実験

4.1 実験目的

本研究の目的は災害などによるインターネット不接続時に、避難所などで行政サービスの申請を行えるようにスマートフォンを用いて申請書類の作成が行えるか確認をする。申請書類を作成するのに専用アプリを必要としないウェブ形式での作製できるシステムを構築し、動作の評価を行う。

4.2 実験環境

実験条件を以下に示す。

表 1 本研究における実験環境

実験場所	自宅
使用機器	Android タブレット WindowsPC
使用ソフト	Apache (2.4.46) MySQL (8.0) PHP (8.0)

4.3 システム構築

本節では実際にシステムが動作する順に沿って構築したシステムの概説を行う。

4.3.1 バンドルファイル(wbn ファイル)

以下の図 4.1 はシステム構築用に作成したり災証明書発行申請用ウェブページである。このウェブページの構成リソースは以下の通りである。

- file_out.html
- test2.jpg

り災証明書申請フォーム

例
 住所 : 滋賀県大津市瀬田大江町 1 - 5
 連絡先 : 080 1234 5678
 氏名 : 龍谷 太郎
 生年月日 : 2000年00月00日
 り災物件との関係 : 居住者 所有者 代理人
 り災年月日 : 2000年00月00日
 災害の種類 : 暴風 竜巻 豪雨 洪水 高潮 地震 津波 噴火
 り災物件 : 住家 非住家 持家 貸家 借家
 り災場所 : 大津市
 り災物件の状況 : 全壊 大規模半壊 半壊 一部損壊

住所 :
 連絡先 :
 氏名 :
 生年月日 : 年 月 日
 り災物件との関係 : 居住者 所有者 代理人
 り災年月日 : 年 月 日
 災害の種類 : 暴風 竜巻 洪水 高潮 地震 津波 噴火
 り災物件 : 住家 非住家 持家 貸家 借家
 り災場所 :
 り災物件の状況 : 全壊 大規模半壊 半壊 一部損壊

ファイル作成

図 4.1 申請フォーム

こちらの申請ページをオフラインでも動作するように Web Bundles を用いてバンドルする必要がある。

4.3.2 申請ファイル(HTML ファイル)

4.3.1 で生成したファイルには JavaScript が埋め込まれており、フォームに入力した状態でページ下部のファイル作成のボタンを押すと入力内容を保持した新たな HTML ファイルが生成される。その様子が以下の図 4.2 である。

り災証明書申請フォーム × テスト用ページ × +

ファイル | /Users/takayamajunakira/Downloads/outFile%20...

Form1 : 滋賀県草津市野路3丁目1-5
 Form2 : 0775690373
 Form3 : 山田 二郎
 Form4 : 20000204
 Form5 : 居住者
 Form6 : 20000204
 Form7 : 地震
 Form8 : 貸家
 Form9 : 草津市
 Form10 : 半壊
 送信

図 4.2 新しく生成された HTML ファイル

図 4.2 に示されている通り、はじめの申請用フォームに記入した内容が既にフォームに記入された状態で新たな HTML ファイルが生成されている。ここで生成されたファイルが申請ファイルとなる。

4.3.3 自己証明書の作成

・秘密鍵(server.kye)の作成

公開鍵暗号方式として RSA を利用するので、RSA 方式の秘密鍵を作成する。作成には openssl genrsa コマンドを利用し、コマンドに鍵長を指定する。今回は 2048bit の秘密鍵を作成する。



```
選択管理者: コマンドプロンプト
Microsoft Windows [Version 10.0.18363.1171]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd c:\Apache24\conf

c:\Apache24\conf>.\bin\openssl.exe genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
+++++
e is 65537 (0x010001)
c:\Apache24\conf>
```

図 4.3 秘密鍵作成

・公開鍵(server.csr)の作成

OpenSSL で公開鍵を作成するのに openssl req コマンドを実行する。

コマンドを実行すると、証明書を発行して欲しいサーバーの情報の入力を促されますので、以下の図 5 のようにサーバーの情報を入力する。


```
選択管理者: コマンド プロンプト - ..\bin\openssl.exe req -new -key server.key -out server.csr -config openssl.cnf
c:\Apache24\conf> .\bin\openssl.exe req -new -key server.key -out server.csr -config openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Siga
Locality Name (eg, city) []:otu
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ryukoku.uni
Organizational Unit Name (eg, section) []:jyoume
Common Name (e.g. server FQDN or YOUR name) []:takayama
Email Address []:t170494@mail.ryukoku.ac.jp

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

図 4.4 公開鍵の作成

・デジタル証明書(server.crt)の作成

最後に成果物であるデジタル証明書を作成します。普通であれば上記で作成した公開鍵(server.csr)を VeriSing などの機関に送付して認証局の秘密鍵で署名してもらいますが、今回は自分で署名することでデジタル証明書作成するので、上記で作成した秘密鍵(server.key)で署名する。デジタル証明書に署名するのに openssl x509 コマンドを実行する。今回は証明書の有効期限が 10 年(3650 日)ある証明書を作成する。

```
管理者: コマンド プロンプト
Microsoft Windows [Version 10.0.18363.1171]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>cd c:\Apache24\conf
c:\Apache24\conf> .\bin\openssl.exe x509 -in server.csr -days 3650 -req -signkey server.key > server.crt
Signature ok
subject=C = JP, ST = Siga, L = ootu, O = ryukoku.uni, OU = jyouhoumedhia, CN = takayama, emailAddress = t170494@mail.ryukoku.ac.jp
Getting Private key
c:\Apache24\conf>
```

図 4.5 デジタル証明書の作成

4.3.4 自己証明書の付与した html ファイルから har ファイル作成

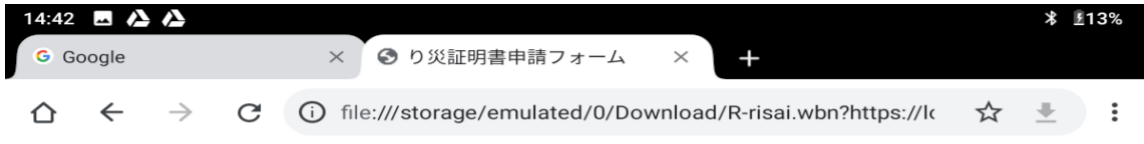


図 4.6 自己証明書を付与した html ファイル

図 4.6 で作成したページを開き、GoogleChrome のデベロッパーツールを開く。そこで har ファイルを作成する。次に作成した har ファイルを wbn ファイルに変換するためにコマンドプロンプトで(例) [gen-bundle -har foo.har -o foo.wbn -primaryURL https://example.com/]と打ち込む。[5]

4.3.5 bluetooth でファイルの転送

作成した wbn ファイルを bluetooth を用いてスマートフォンに転送する。転送されたファイルはスマートフォンアプリの「Files」にダウンロードされる。Files 内の wbn ファイルが保存されているリンクのコピーを行う。次に GoogleChrome アプリを開きコピーしたリンクを貼り付けて図 8 のように展開する。



り災証明書申請フォーム

例
住所 : 滋賀県大津市瀬田大江町1-5
連絡先 : 080 1234 5678
氏名 : 龍谷 太郎
生年月日 : 2000年00月00日
り災物件との関係 : 居住者 所有者 代理人
り災年月日 : 2000年00月00日
災害の種類 : 暴風 竜巻 豪雨 洪水 高潮 地震 津波 噴火
り災物件 : 住家 非住家 持家 貸家 借家
り災場所 : 大津市
り災物件の状況 : 全壊 大規模半壊 半壊 一部損壊

住所	: 住所を入力してください
連絡先	: [] - [] - []
氏名	: 氏名を入力してください
生年月日	: 0000 年 00 月 00 日
り災物件との関係	: <input type="radio"/> 居住者 <input type="radio"/> 所有者 <input type="radio"/> 代理人
り災年月日	: 0000 年 00 月 00 日
災害の種類	: <input type="radio"/> 暴風 <input type="radio"/> 竜巻 <input type="radio"/> 洪水 <input type="radio"/> 高潮 <input type="radio"/> 地震 <input type="radio"/> 津波 <input type="radio"/> 噴火
り災物件	: <input type="radio"/> 住家 <input type="radio"/> 非住家 <input type="radio"/> 持家 <input type="radio"/> 貸家 <input type="radio"/> 借家
り災場所	: []市
り災物件の状況	: <input type="radio"/> 全壊 <input type="radio"/> 大規模半壊 <input type="radio"/> 半壊 <input type="radio"/> 一部損壊

[ファイル作成](#)

図 4.7 スマートフォンで展開した wbn ファイル

申請フォームを開いたら 4.3.2 のようにデータを入力します。データの入力が終わると次にページ下部にある「ファイル作成」を押す。入力されたデータから新しく html ファイルが生成される。新しく生成された html ファイルを再び bluetooth を用いて PC に転送を行う。

4.3.6 データベースへ格納

PC に転送された html ファイルは図 4.2 のような html ファイルが届くと、ページ下部の「送信」をクリックするとデータベースに格納される。

```

mysql> show create table rwbm;
+-----+-----+
| Table | Create Table |
+-----+-----+
| rwbm | CREATE TABLE `rwbm` (
  `address` text,
  `tel` text,
  `name` text,
  `date` date DEFAULT NULL,
  `property_relation` text,
  `D_date` date DEFAULT NULL,
  `D_type` text,
  `D_property` text,
  `D_place` text,
  `status` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
| row in set (0.00 sec) |
mysql> select * from rwbm;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| address | tel | name | date | property_relation | D_date | D_type | D_property | D_place | status |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 滋賀県大津市一里山6丁目9-1 | 0775452811 | 瀬田 一郎 | 2000-10-12 | 所有者 | 2000-10-12 | 暴風 | 持家 | 大津市 | 大規模半 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| row in set (0.00 sec) |
mysql>

```

図 4.8 データベースに入力されたデータを格納

4.4 考察

本研究の実験を通して、オフライン環境で web ベースの申請内容を避難所から役所へ運搬し、データベースに格納するまでの流れが実現できた。実験の際、Android OS では wbn ファイルが利用可能だが、iOS では wbn ファイルの展開が不可能であった。また wbn ファイルを展開するのに現状 GoogleChrome でしか運用できず、拡張機能の設定が必要である。実用化されるまでには多くの課題があると考えられる。wbn ファイルを作成するまでに自己証明証を発行するまでに非常に手間がかかるが、SSL 証明書が付与されている web ページを利用すれば wbn ファイルはすぐに作成できた可能性がある。各市町村で、り災証明書を web で記入できるページを作成しておけば、災害発生時にすぐに wbn ファイルとして活用できるのではないかと考える。

第5章 結論

本論文では、web bundles を用いた行政システムのオフライン電子申請システムの実装について述べた。web bundles を用いることで、オフライン環境で web ページを動作させ、データ収集することも実現した。マイナンバーカードによる電子署名が実現できれば、誤った内容の申請や不正に申請することを防ぐ事ができる。マイナンバーカードの電子署名の有効性が今後の課題となる。

謝辞

本研究課題を与えていただき、終始ご指導、ご鞭撻を賜りました三好 力 教授に深く感謝の意を表します。

参考文献

- [1] Get started with Web Bundles(Nov11,2019)
<https://web.dev/web-bundles/>
- [2] くるくらの災害で避難する際に持っていくものは(2018年9月27日)
<https://kurukura.jp/news/180926-10.html>
- [3] TIME&SPACEbyKDDI bluetooth のバージョン
<https://time-space.kddi.com/ict-keywords/20190909/2738>
- [4] 山村新也 , 永田晃 : DTN:遅延と仲良くするネットワーク , 通信ソサイエティマガジン , No.16[春号] , pp.52-56(2011)
- [5] WICG/webpackage/go/bundle
<https://github.com/WICG/webpackage/tree/master/go/bundle>
- [6] PWA とはメリットと実装事例について
<https://digital-marketing.jp/seo/what-is-progressive-web-apps/>
- [7] 内閣府 : 災害にまつわる住家の被害認定:防災情報のページ
<http://www.bousai.go.jp/taisaku/unyou.html>
- [8] OYO Hotels Japan :
<https://www.oyorooms.com/jp/>

付録

り災証明書申請フォームソースコード

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>り災証明書申請フォーム</title>
  <h1>り災証明書申請フォーム</h1>
</head>
<script>
  function dispText() {
    var text = '<head>\n'+
      '<meta charset="UTF-8">\n'+
      '<title>テスト用ページ</title>\n'+
      '</head>\n'+
      '<body>\n'+
      '<form name="form" id="id" action="form_check.php" method="post">\n'+
      'Form1 :<input type="text" name="a001" size="50" value="" +
      document.formname.a001.value +
      ">\n</br>\n'+
      'Form2 :<input type="text" name="a002" size="12" value="" +
      document.formname.b021.value +
      document.formname.b022.value +
      document.formname.b023.value +
      ">\n</br>\n'+
      'Form3 :<input type="text" name="a003" size="20" value="" +
      document.formname.a003.value +
      ">\n</br>\n'+
      'Form4 :<input type="text" name="a004" size="8" value="" +
      document.formname.b041.value +
      document.formname.b042.value +
      document.formname.b043.value +
      ">\n</br>\n'+
      'Form5 :<input type="text" name="a005" size="30" value="" +
      document.formname.a005.value +
      ">\n</br>\n'+
      'Form6 :<input type="text" name="a006" size="8" value="" +
      document.formname.b041.value +
      document.formname.b042.value +
      document.formname.b043.value +
      ">\n</br>\n'+
      'Form7 :<input type="text" name="a007" size="4" value="" +
      document.formname.a007.value +
      ">\n</br>\n'+
      'Form8 :<input type="text" name="a008" size="4" value="" +
      document.formname.a008.value +
      ">\n</br>\n'+
      'Form9 :<input type="text" name="a009" size="6" value="" +
      document.formname.a009.value +
      ">\n</br>\n'+
      'Form10:<input type="text" name="a010" size="6" value="" +
      document.formname.a010.value +
      ">\n</br>\n'+
      '<input type="submit" value="送信">'
    '</body>\n'+
    '</html>';

    var blob = new Blob([text], { "type": "text/plain" });

    //IE の場合
    if (window.navigator.msSaveBlob) {
      window.navigator.msSaveBlob(blob, "outFileFromWindows.txt");
    }
    //IE 以外の場合
    else {
      document.getElementById("createFile").href = window.URL.createObjectURL(blob);
    }
  }
</script>
</head>
<body>
  

  <form name="formname" id="id_form" action="">
  <fieldset>
  <label id="001">住所 :</label>
  <input required type="text" name="a001" size="50" maxlength="50" required placeholder="住所を入力してください">
  </br>

  <label id="002">連絡先 :</label>
  <input type="text" name="b021" size="4" maxlength="4" value="" > -
  <input type="text" name="b022" size="4" maxlength="4" value="" > -
  <input type="text" name="b023" size="4" maxlength="4" value="" >
  </br>

  <label id="003">氏名 :</label>
  <input type="text" name="a003" size="30" maxlength="30" placeholder="氏名を入力してください">
  </br>
  <label id="004">生年月日 :</label>
  <input type="text" name="b041" size="4" maxlength="4" placeholder="0000">
  年
  <input type="text" name="b042" size="2" maxlength="2" placeholder="00">
  月
  <input type="text" name="b043" size="2" maxlength="2" placeholder="00">
  日
  </br>
  <label id="005">り災物件との関係 :</label>
  <input type="radio" name="a005" value="居住者">居住者
  <input type="radio" name="a005" value="所有者">所有者
  <input type="radio" name="a005" value="代理人">代理人
  </form>
</body>
</html>
```



```

</br>
<label id="006">り災年月日      :</label>
<input type="text" name="b061" size="4" maxlenglabel="4" placeholder="0000">
年
<input type="text" name="b062" size="2" maxlenglabel="2" placeholder="00">
月
<input type="text" name="b063" size="2" maxlenglabel="2" placeholder="00">
日
</br>
<label id="007">災害の種類      :</label>
<input type="radio" name="a007" value="暴風">暴風
<input type="radio" name="a007" value="竜巻">竜巻
<input type="radio" name="a007" value="洪水">洪水
<input type="radio" name="a007" value="高潮">高潮
<input type="radio" name="a007" value="地震">地震
<input type="radio" name="a007" value="津波">津波
<input type="radio" name="a007" value="噴火">噴火
<!--
<input type="radio" name="a007" value="">その他
<input type="hidden" value="">
<span class="msg"></span>
その他に入力欄を追加する
<textarea name="a007" rows="1" wrap="soft" ></textarea><br>
-->

</br>
<label id="008">り災物件      :</label>
<input type="radio" name="a008" value="住家">住家
<input type="radio" name="a008" value="非住家">非住家
<input type="radio" name="a008" value="持家">持家
<input type="radio" name="a008" value="賃家">賃家
<input type="radio" name="a008" value="借家">借家
</br>
<label id="009">り災場所      :</label>
<input type="text" name="a009" size="10" maxlenglabel="10" placeholder="〇〇市">
</br>
<label id="010">り災物件の状況 :</label>
<input type="radio" name="a010" value="全壊">全壊
<input type="radio" name="a010" value="大規模半壊">大規模半壊
<input type="radio" name="a010" value="半壊">半壊
<input type="radio" name="a010" value="一部損壊">一部損壊
</fieldset>
</br>
</br>
</br>
<a id="createFile" href="#" download="outFile.html" value="ファイル作成" onclick="dispText()">ファイル作成</a>
</form>
</body>
</html>

```

データベースにデータを格納する PHP ファイル

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>情報入力</title>
</head>
<body>
<?php

$con = mysqli_connect('127.0.0.1', 'root', 'rwbntest');
if (!$con){
    exit('データベースに接続できませんでした。');
}

$result = mysqli_select_db($con, 'test2');
if (!$result){
    exit('データベースを選択できませんでした。');
}

$result = mysqli_query($con, 'SET NAMES utf8');
if (!$result){
    exit('文字コードを指定できませんでした。');
}

$address      = $_REQUEST['a001'];
$tel          = $_REQUEST['a002'];
$name         = $_REQUEST['a003'];
$date        = $_REQUEST['a004'];
$property_relation = $_REQUEST['a005'];
$D_date      = $_REQUEST['a006'];
$D_type      = $_REQUEST['a007'];
$D_property  = $_REQUEST['a008'];
$D_place     = $_REQUEST['a009'];
$status      = $_REQUEST['a010'];

$result = mysqli_query($con, "INSERT INTO Rwbn VALUES('$address','$tel','$name','$date','$property_relation','$D_date','$D_type','$D_property','$D_place','$status')");
if (!$result){
    exit('データを登録できませんでした。');
}

$con = mysqli_close($con);
if (!$con){
    exit('データベースとの接続を閉じられませんでした。');
}

?>
<p>登録が完了しました。<br /><a href="index.html">戻る</a></p>
</body>
</html>

```