

卒業研究報告

題目

サウンドコード技術を利用したIoTセキュリティシステムのソフトウェア開発

指導教官

高橋寛教授

王森レイ講師

報告者

井早千裕

平成 30 年～月～日提出

愛媛大学工学部情報工学科計算機システム工学講座

目次

第 1 章	まえがき	1
1.1	研究背景および目的・目標	1
1.2	本論文の構成	2
第 2 章	準備	3
2.1	諸定義および用語説明	3
第 3 章	サウンドコード技術について	6
3.1	サウンドコード技術の概要	6
第 4 章	サウンドコード技術を利用した宅配ボックスシステムについて	8
4.1	サウンドコード技術を利用した宅配ボックスシステム	8
第 5 章	研究結果・考察	13
5.1	宅配ボックスシステム開発について	13
5.2	研究結果	14
5.2.1.	UML 作成	14
5.2.2.	ソフトウェア開発	21
5.2.3.	テスト	33
5.3	考察	41
第 6 章	あとがき	43
	謝辞	44

参考文献

45

第 1 章

まえがき

1.1 研究背景および目的・目標

サウンドコードは文字情報を音に変換して送信する新しい音声通信技術である [4]. その特徴は, 特別な機器を必要としないため導入コストや消費電力が低く, 人には聞こえない音を使用しているため機密性が高いというものである [4]. しかしながら, その応用はまだ進んでいない. そこで, サウンドコードの機密性などの特徴を IoT セキュリティシステムに応用することを提案する. IoT セキュリティシステムにはホームセキュリティシステムなど様々な例があるが, 本研究では, IoT 化した宅配ボックスシステムへのサウンドコードの利用を提案する.

宅配ボックスシステムへサウンドコードを応用する理由として宅急便の再配達問題を挙げる. 近年の電子商取引 (EC) 市場の拡大に伴い, 宅急便の取引個数が急激に増加している [1]. その中で, 宅配貨物の不在再配達は全体の約 2 割発生している [1]. この再配達を労働力に換算すると, 年間約 9 万人のドライバーの労働力に相当し, また, 再配達を行うトラックから排出される二酸化炭素の量は年間でおおよそ 42 万トンとなる [1]. この問題は労働生産性の低下や, 地球温暖化の増進につながるため, 再配達を減らす必要がある.

現在の再配達削減に向けた消費者の取組として, 内閣府が平成 29 年に発表した「再配達問題に関する世論調査」では, 「鉄道駅等の公共スペースに設置された, 誰でも利用可能な宅配ロッカーへの配達」が 0.7% と最も低い結果であった. 利用しない理由とし

てセキュリティに不安があると答えた方が 19.2%に及び、宅配ボックスのセキュリティ向上の必要性があることが考えられる [2]。一方で「宅配事業者や通販事業者の WEB、アプリ等を活用した配達日時や受取場所 の指定・変更」を挙げた方が 14.5%と最も高く、スマートフォンアプリケーションなどで宅配ボックスの利用管理するなど宅配ボックスを IoT 化する有効性がうかがえる [2]。

そこで本研究では、サウンドコード技術を利用した IoT 宅配ボックスシステムを開発することで、サウンドコード技術をセキュリティシステムに利用する問題点や有効性を確認することを目的とする。また、V 字開発モデルに従って設計・コーディング・テストを経て開発を行い、宅配ボックスシステムの評価を行うことを目標とする。なお、本研究は、グループで V 字開発モデルに従って組込みシステムを開発する

本研究では、要件定理・テスト・役割分担の容易化のために、UML の 3 つの図を使用し設計を行った。それを基に研究メンバーで役割分担を行った。研究メンバーのタスク管理とプロジェクト管理のため、ガントチャートを作成し、それに沿って開発を行った。サウンドコードのエンコーダー、デコーダー、は Raspberry Pi で行うため Raspberry Pi 上にアプリケーションを開発した。

1.2 本論文の構成

本論文は以下のような構成となっている。第 1 章は、本研究に至った研究背景および目的、目標について述べる。第 2 章では、本論文に必要な用語の説明について述べる。第 3 章では、本研究に用いたサウンドコード技術についておよびサウンドコードを本研究に使用する理由を説明する。第 4 章では、本研究で開発を行ったサウンドコード技術を利用した宅配ボックスシステムについて説明する。第 5 章は実施した実験方法および結果考察について述べる。第 6 章は、本論文のまとめと今後の課題について述べる。

第 2 章

準備

本章では, 本研究で用いた言語について説明する.

2.1 諸定義および用語説明

V 字開発モデル

V 字開発モデルとは, システム開発の設計から実装・テストまでの一連の流れにおける, 開発工程とテスト工程の対応関係を表したモデルである [5]. V 字開発モデルを図 2.1 に示す. 図 2.1 の横軸は開発の時間軸であり, 縦軸は詳細化の程度を表している [6]. V 字開発モデルのプロセスでは, V 字の左側で要件定義, 基本設計, 詳細設計と開発工程の上流工程から順番に右下がりに行い, V 字の頂点における部分で実装 (コーディング) を行う. そして V 字の右側で, 単体テスト, 結合テスト, 総合テストと左側に並べた開発工程に対応したテスト工程を右上がりに行う. このように V 字開発モデルに従い開発・テストを行うことで, プロジェクトの進行段階が容易に把握できるとともに, 実施されるテストがどのレベルの開発内容を検証するためのテストなのか, 何に着目したテストが行われるのかを示すことができる [5].

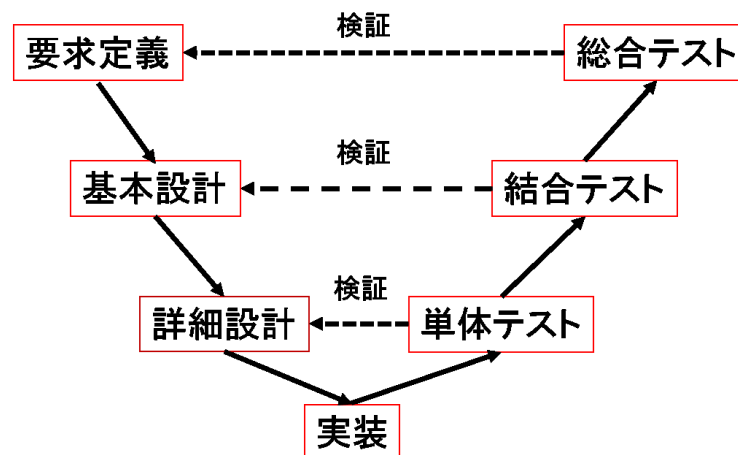


図 2.1. V字開発モデル

UML

統一モデリング言語（Unified Modeling Language）の略称であり、要件定義から実装、テストまでの各局面で利用するダイアグラムとその表記法を統一したモデリング言語である。主なものに、ユースケース図、シーケンス図、クラス図などがある [3]。

ユースケース図

ユーザやクライアントの要求事項の表現および、システムに対して課せられている基本機能やサービス項目などの要件定義の表現に使用される図である。現在考えているシステムを中心に置き、アクター（システムとその利用者（外部システムを含む））とのやり取りをユースケースとして書いていく [3]。

シーケンス図

時間経過の中でどう動くかという動的なふるまいを表現する図である。上から下に時間が経過するという前提で、その時間に沿って、ある機能やふるまいを実現するのに必要なメッセージのやり取りを書いていく [3]。

クラス図

オブジェクト指向分析設計における中心図で、モデルの静的な構造の表現に使用

される [3].

宅配ボックスシステム

サウンドコード技術を利用した宅配ボックスシステムを省略したもの.

エンコーダー

サウンドコードを生成する実行プログラムである.

デコーダー

サウンドコードを解析する実行プログラムである.

単体テスト

本研究では宅配ボックスシステムを一つのシステムと考え, アプリ単体や, サーバ単体, ボックス単体のテストのことを単体テストとする. モジュール単位ではない.

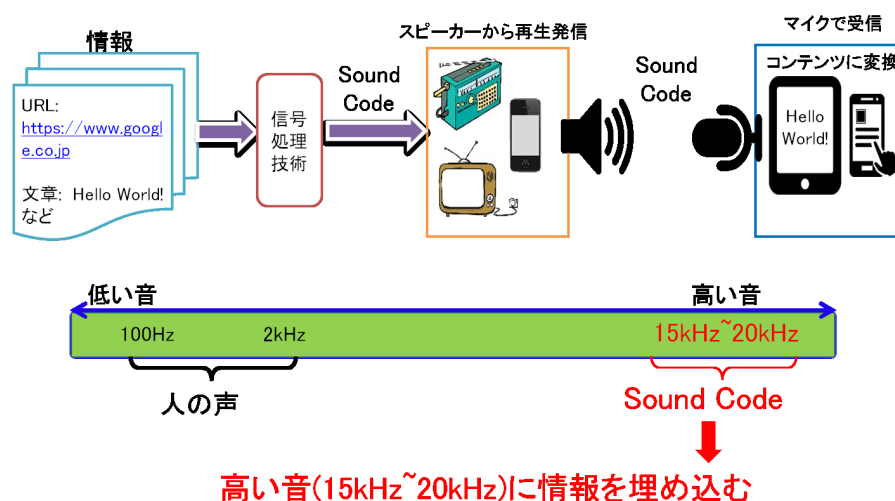
第 3 章

サウンドコード技術について

本章では, サウンドコード技術について説明する. また, サウンドコード技術を IoT システムに利用する理由を説明する.

3.1 サウンドコード技術の概要

サウンドコードとは, URL やライセンスキーなどの文字情報を, 音に変換して送信する音声通信技術のことである. 波形認識ではなく, 音声によるデータ通信技術である [4].



サウンドコードの特徴は次のとおりである.

- テレビやスマートフォンなどの身近なスピーカーを使用できる
- 人には聞こえない音を使用しており, ほかの音声に重畳することも可能
- インターネット環境不要
- サウンドコードの生成・解析が容易
- 低コストの CPU で管理可能で, 低消費電力

以上の特徴から分かるように, サウンドコードは人に聞こえない音であることから, 機密性が高く, 秘密鍵として使用しても, 従来のセキュリティシステムのように安易に暗証番号が漏れたり, 鍵やカードの紛失により安全性を失われることがない. また, サウンドコードの生成が容易であることから, 秘密鍵の更新が容易となり, 管理コストの低減や, 録音対策にもつながる. また, 専用機器は必要なく, 身近なスピーカーとマイクがあれば利用可能であるため導入コストが少ない. また, 文字データを音声に変換する技術であるため, 鍵だけでなく, 荷物の預け入れ状態や受領書などのやり取りもできる. このことから, サウンドコードを秘密鍵として利用するメリットがある.

今回使用するサウンドコードのエンコードはエンコーダーを利用する. エンコーダーを実行する際, 最低限 wav ファイル名とユニーク ID を指定することで wav ファイルが生成される. そこにライセンスキー (16 進数 8 ケタ \times 4) を付加することができ, 本研究ではライセンスキーに情報を入れ込むこととした. また, デコードは, デコーダーを利用する. デコーダーを実行する際, デコードする wav ファイルの wav ファイル名と, ライセンスキーが値を持っていればライセンスキーを指定する必要がある.

第 4 章

サウンドコード技術を利用した宅配ボックスシステムについて

本章では, サウンドコードを利用した宅配ボックスシステムについて説明する. このシステムの中で本研究ではスマートフォンアプリケーションソフトウェアを開発したため, 本章の後半ではスマートフォンアプリケーションについて重点的に説明する.

4.1 サウンドコード技術を利用した宅配ボックスシステム

サウンドコード技術を利用した宅配ボックスシステムは, スマートフォンアプリケーションを使用し, 宅配ボックスの開扉を行うシステムである. その際, 利用者の情報や, ボックスの荷物有無状態などをサーバで管理する. 宅配ボックスシステムでは, サウンドコードをロックの解除, 利用登録, 受領書受信, 宅配ボックスの状態受信など, 宅配ボックスとアプリケーションのやり取りに利用する. 開発対象は, 宅配ボックス・利用者向けアプリ・宅配業者向けアプリ・サーバである. 宅配ボックスシステムの模式図を図 4.1 に示す.

以下開発対象の特徴や役割についてそれぞれ述べる.

宅配ボックス

荷物を預け入れる箱である. データベースを持っており, 自分自身のボックス情報テーブル (ボックス番号, シリアルナンバー, シリアルコード, 預け入れ状態) と, 宅

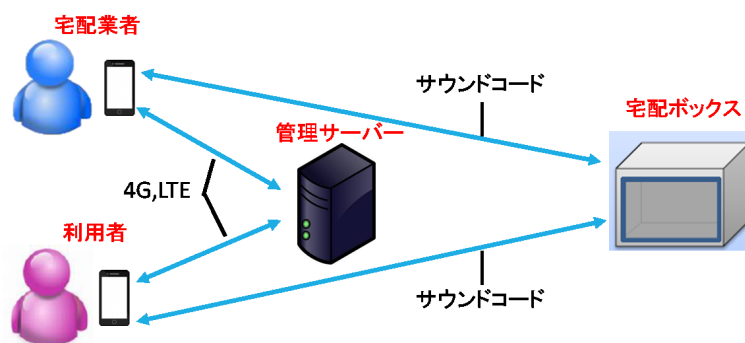


図 4.1. 宅配ボックスシステムの模式図

配業者の情報テーブル (アクセスコード) を持っている. スマートフォンアプリとサウンドコードで通信を行い, サーバとは直接通信を行わない. 宅配ボックスの部品とその役割を以下の表 4.1 に示す.

表 4.1. 宅配ボックスの部品とその役割

項目	説明
Raspberry Pi	サウンドコードを搭載した制御基盤. サウンドコードの送受信と宅配ボックスの制御, データベース管理を行う.
マイク	スマートフォンアプリから送信されるサウンドコードを受信する.
スピーカー	ボックスの預け入れ状態や, 荷物の受領書をサウンドコードでスマートフォンアプリへ送る.
状態表示 LED	荷物の預け入れ状態を表示します. 点灯中は, 荷物が入っているため使用不可であることを表す.
電源ケーブル	電力供給のケーブル.
シリアルナンバー	宅配ボックスの識別番号. 宅配ボックスには, それぞれ固有のシリアルナンバーが Raspberry Pi のデータベースに記録されている.
ボックス番号	宅配ボックスの各ボックス (ドア) の識別番号. それぞれ固有のボックス番号が Raspberry Pi のデータベースに記録されている. また, ボックス番号は, 各ドアに貼付されている.

サーバ

宅配ボックスを管理するサーバである. データベースを持っており, ボックス情報テーブル (ボックス番号, シリアルナンバー, シリアルコード, 預け入れ状態) と受領書情報テーブル (伝票番号, 預け入れ日時, ボックス番号, シリアルナンバー), 宅配業者向けアプリ情報テーブル (アクセスコード, 業者名, 社員番号) を持っている. 主にスマートフォンアプリから送られてくる情報を保管し, スマートフォンアプリからの検索や, 通知機能などに利用する.

利用者向けアプリ

宅配ボックスの利用者が使用するスマートフォンアプリである. 利用者の利用登録, 宅配ボックスの利用者が変更された場合の再利用登録 (シリアルコードの変

更), 荷物の取り出し, 預け入れ通知, サーバへのデータ送信に使用される。

宅配業者向けアプリ

宅配ボックスを利用する宅配業者が使用するスマートフォンアプリである。宅配業者の利用登録, 預け入れ, 受領書取得, 受領書確認, 預け入れ可否確認, サーバへのデータ送信に使用される。

次に宅配ボックスシステムの預け入れフローについて述べる。

1. 宅配業者が宅配業者向けアプリを立ち上げ, 宅配先のボックス番号を入力し, 預け入れボタンを押す。
2. 開扉コードが作成され, サウンドコードによって宅配ボックスへ送信される。
3. 宅配ボックスはサウンドコードを受信すると, ロックを解除しドアを開く。このとき, 荷物がすでに入れている場合はロックを解除しない。
4. 荷物を入れてドアを閉めるとロックがかかり LED が点灯する。
5. 宅配ボックスがサウンドコードで受領書を送信し, 宅配業者向けアプリが受け取る。
6. 宅配業者向けアプリが預け入れ状態と受領書情報をサーバへ php で送信する。

次に宅配ボックスシステムの取り出しフローについて述べる。

1. 利用者が利用者向けアプリを立ち上げ, 取り出しボタンを押す。
2. 解錠コードが作成され, サウンドコードによって宅配ボックスへ送信される。
3. 宅配ボックスはサウンドコードを受信すると, サウンドコードの正当性を検証しロックを解除する。
4. 利用者は荷物を取り出しドアを閉めると, 宅配ボックスはサウンドコードで預け入れ状態を送信し, 利用者向けアプリが受け取る。
5. 利用者向けアプリが預け入れ状態をサーバへ php で送信する。

次に利用者向けアプリについて述べる。

宅配ボックスの利用者は利用者向けのアプリを使い宅配ボックスの開扉を行う。その際利用登録が必要である。利用登録にはシリアルコード・シリアルナンバー・ボックス番号が必要である。シリアルコードとは、宅配ボックスのシリアルナンバーと紐づけられていてセットアップを行うための重要な情報である。そのため、宅配ボックスを利用する人以外の人に知られてはならない。また、同じシリアルコードを使って他の宅配ボックスをセットアップすることはできない。シリアルコードの流出や、宅配ボックスの利用者が変更になった場合は、宅配ボックス管理者がシリアルコードの再登録を行う。シリアルコードの再登録には、シリアルナンバー・ボックス番号・新しいシリアルコード・マスターシリアルコードが必要である。マスターシリアルコードとは、宅配ボックス管理者のみが持っているもので、マスターシリアルコードがないとシリアルコードの再登録はできない。宅配ボックス管理者は、新しいシリアルコードを設定すると、そのシリアルコードを宅配ボックス利用者に伝える。利用者向けアプリのホーム画面には預け入れ状態が保存されているサーバから預け入れ状態が送られ通知される。

本研究ではデコーダ・エンコーダがボードコンピュータの Raspberry Pi 上で動くものとなっているため、利用者向けアプリは Raspberry Pi 上で php を利用して開発することとした。

次に宅配業者向けアプリについて述べる。

宅配ボックスに荷物を届ける宅配業者は、宅配業者向けアプリを使い宅配ボックスの開扉を行う。その際利用登録が必要である。利用登録には、業者名、社員番号、アクセスコードが必要となる。アクセスコードとはアプリごとに割り当てられるコードである。受領書確認のために必要である。宅配業者向けアプリでは、宅配するボックスのボックス番号とシリアルナンバーを入力し、サーバに送信すると、そのボックスに預け入れ可能かどうか（預け入れたい宅配ボックスにすでに荷物が預け入れられていた場合預け入れできないため）検索できるようになっている。また、受領書確認ボタンを押すことで、アクセスコードで受領書がサーバから検索され、その結果が表示される。

本研究ではデコーダ・エンコーダがボードコンピュータの Raspberry Pi 上で動くものとなっているため、宅配業者向けアプリは Raspberry Pi 上で php を利用して開発することとした。

第 5 章

研究結果・考察

5.1 宅配ボックスシステム開発について

本章では、本研究で実施した宅配ボックスシステム開発について述べる。

本研究では、初めに研究メンバーと共に UML を使用し、システム設計を行った。使用した図は、ユースケース図・シーケンス図・クラス図である。次に、作成したクラス図を基に役割分担を行い、研究メンバー共通のガントチャートに各自タスクを記入した。使用したガントチャート作成ツールは「Brabio!」である。作成したガントチャートを図 5.1 に示す。



図 5.1. ガントチャート

次に、作成した図に従い利用者向けアプリ・宅配業者向けアプリを開発した。開発後、単体テストを行い、問題修正を行ったのち宅配ボックスとサーバとの結合テストを行った。

5.2 研究結果

ここでは研究結果を UML・ソフトウェア開発・テストの 3つのカテゴリに分けて述べる.

5.2.1. UML 作成

研究環境 以下に研究環境を示す.

OS	Windows8.1
使用ソフト	astah community

研究方法 以下に研究方法を示す.

1. astah community を使用し, 各自 UML を作成する.
2. 各自が作成した UML を統合する.
3. 修正があれば, 修正を行う.

研究結果 以下に研究結果を示す.

ユースケース図を図 5.2 に示す.

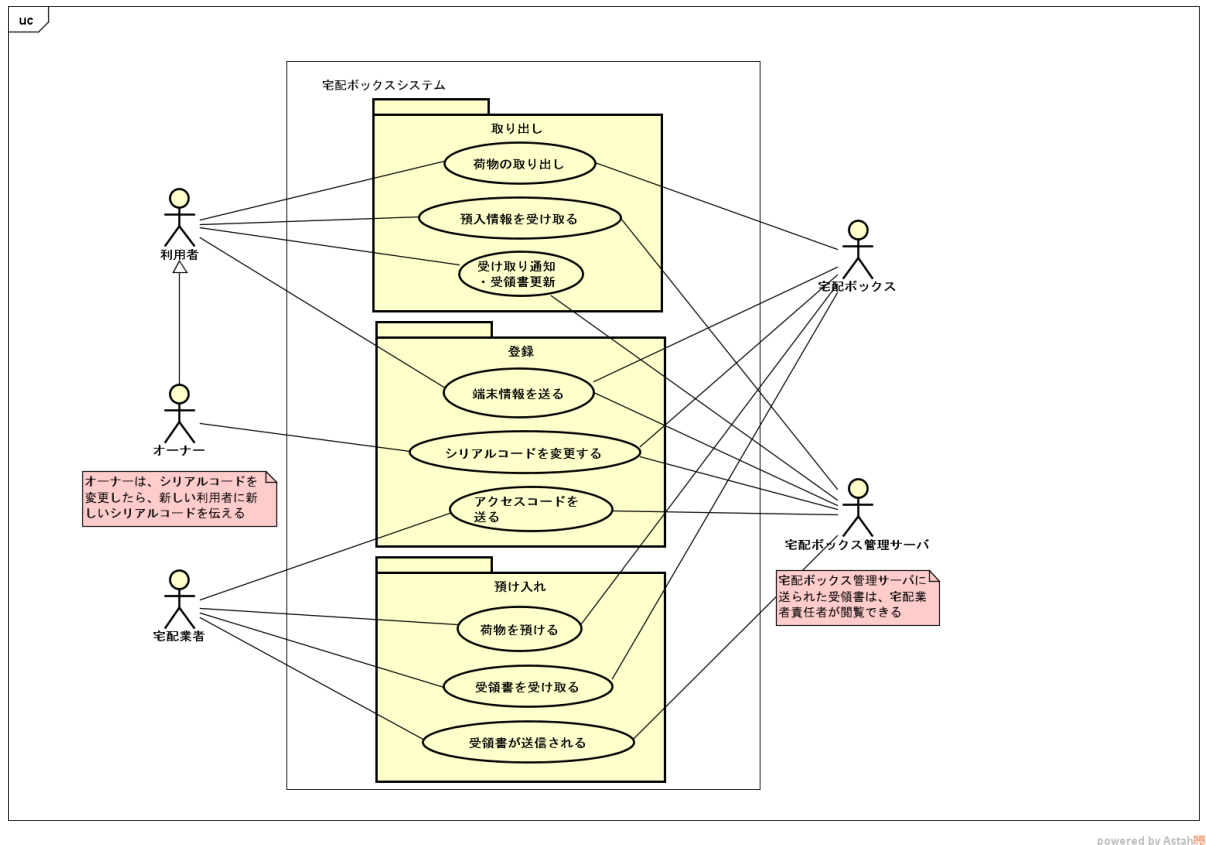


図 5.2. ユースケース図

ユースケース図では, 利用者や宅配業者視点での, 宅配ボックスシステムの機能を表現している. シーケンス図によって宅配ボックスシステムの機能の洗い出しや, 「利用者・宅配業者」と「宅配ボックス・サーバ」との機能ごとのつながりを表現することにより, 要求定義を行う.

シーケンス図を図 5.3 と 5.4 に示す.

シーケンス図では, ユースケース図で定義した利用者・宅配業者・宅配ボックス・サーバというアクターに加え, 開発対象である利用者向けアプリ・宅配業者向けアプリの 6 つのアクター間での動的なふるまいを時間経過に沿って表している. 青い枠で囲まれた opt という部分はオプションと言い, [] 内に記述されている条件を満たした場合のみ実行される処理である [7]. オプションは主に, 登録の機能を記述する際に用いた. オレンジ・ピンク・水色・黄緑色の枠で囲まれた alt という部分はオルタナティブと言い, [] 内に記述された条件での分岐処理を表している [7]. 枠は破線で 2 つに区切られており, 上半分には条件が正しい場合のふるまいを記述し, 下半分では条件が誤りである場合のふるまいを記述する. 主に開扉コードの正当性・扉の開閉状態・荷物有無状態の分岐で使用している. 基本設計として利用している. このシーケンス図を作成する段階でサウンドコードを録音された場合のことを考慮し, 開扉回数を数え, その情報をサウンドコードに付加することで宅配ボックスを一回開扉するごとにサウンドコードを更新するという対策を考え記述している.

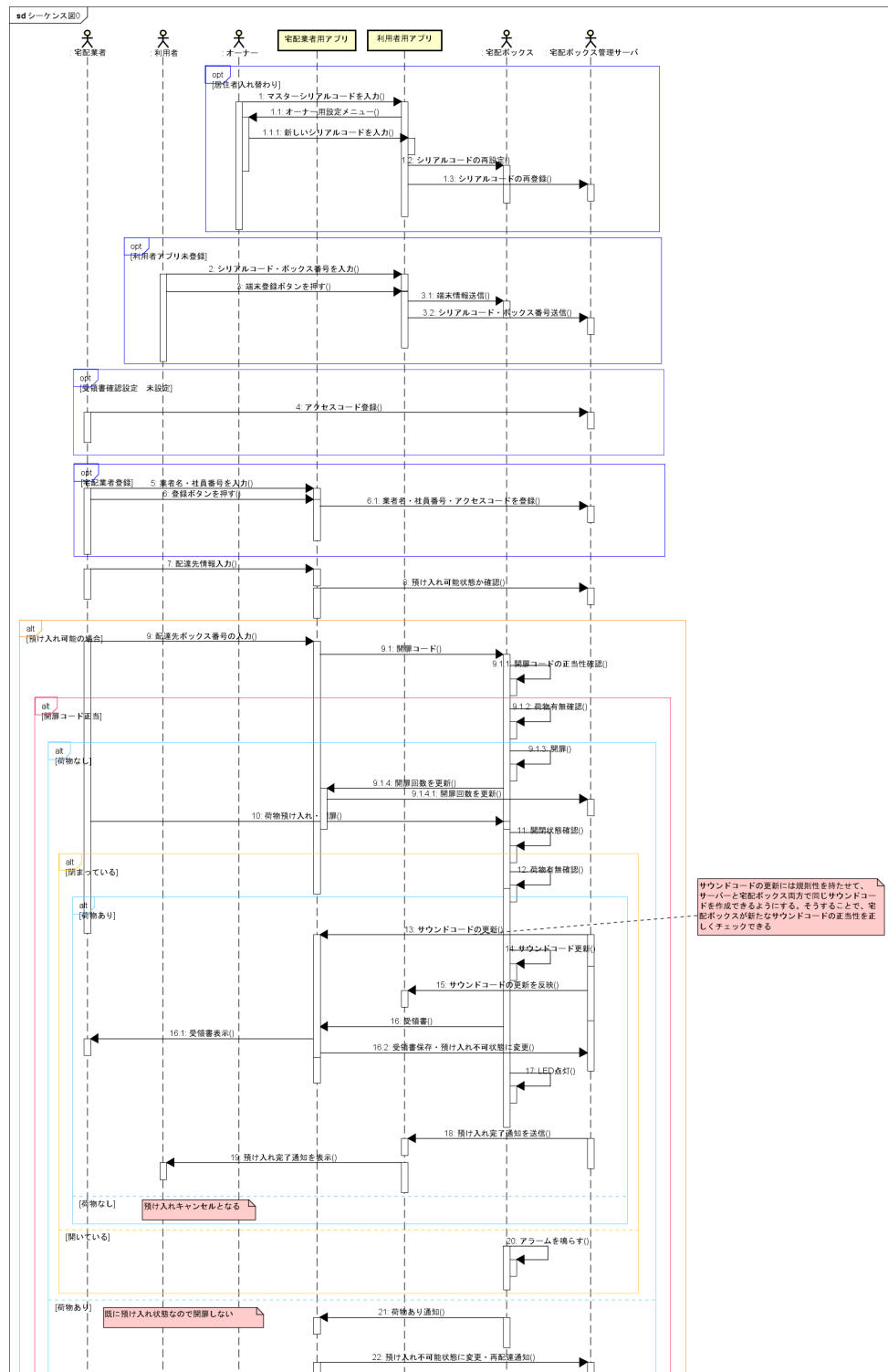


図 5.3. シーケンス図(上半分)

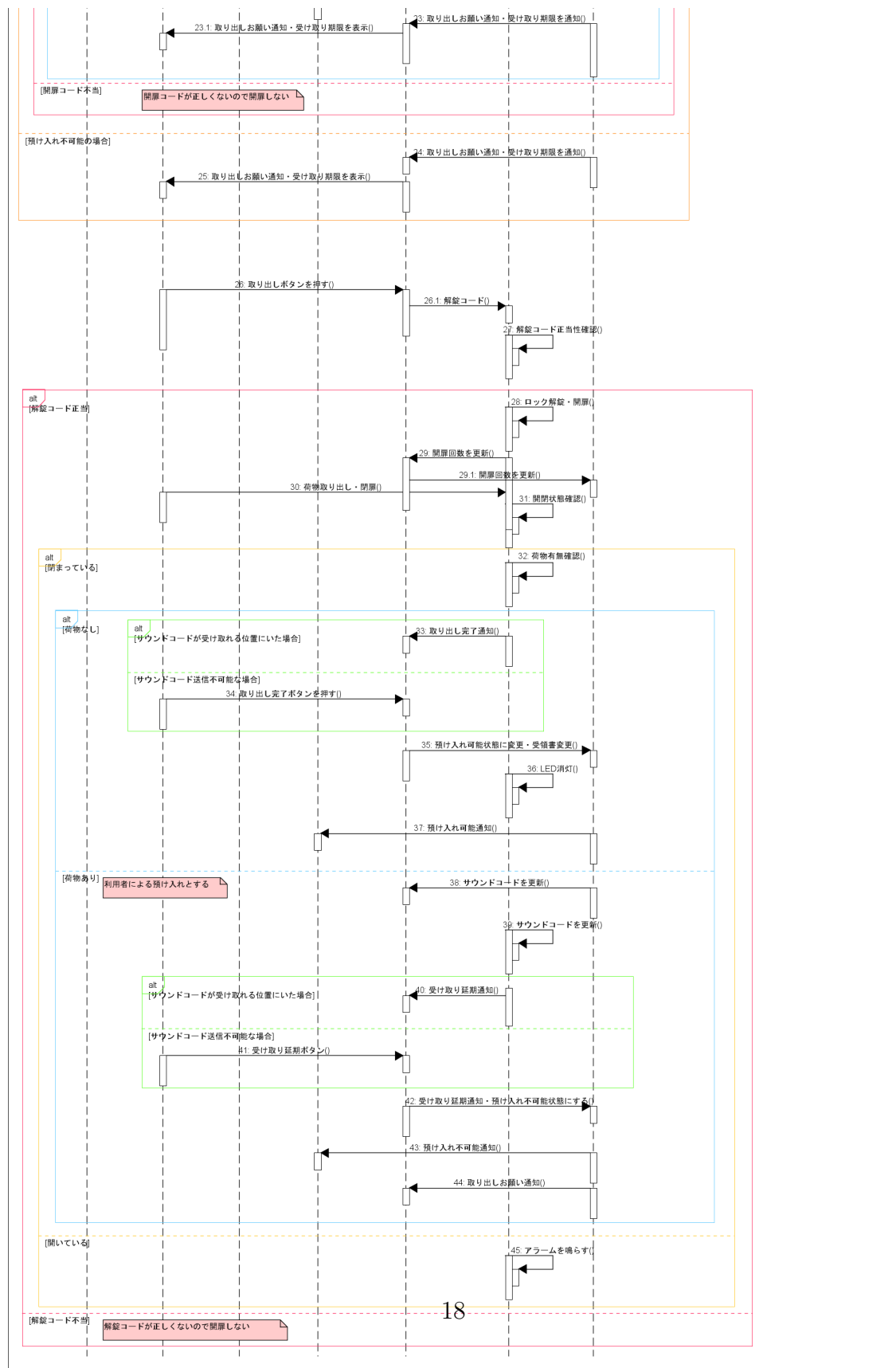


図 5.4. シーケンス図 (下半分)

クラス図を図 5.5 に示す.

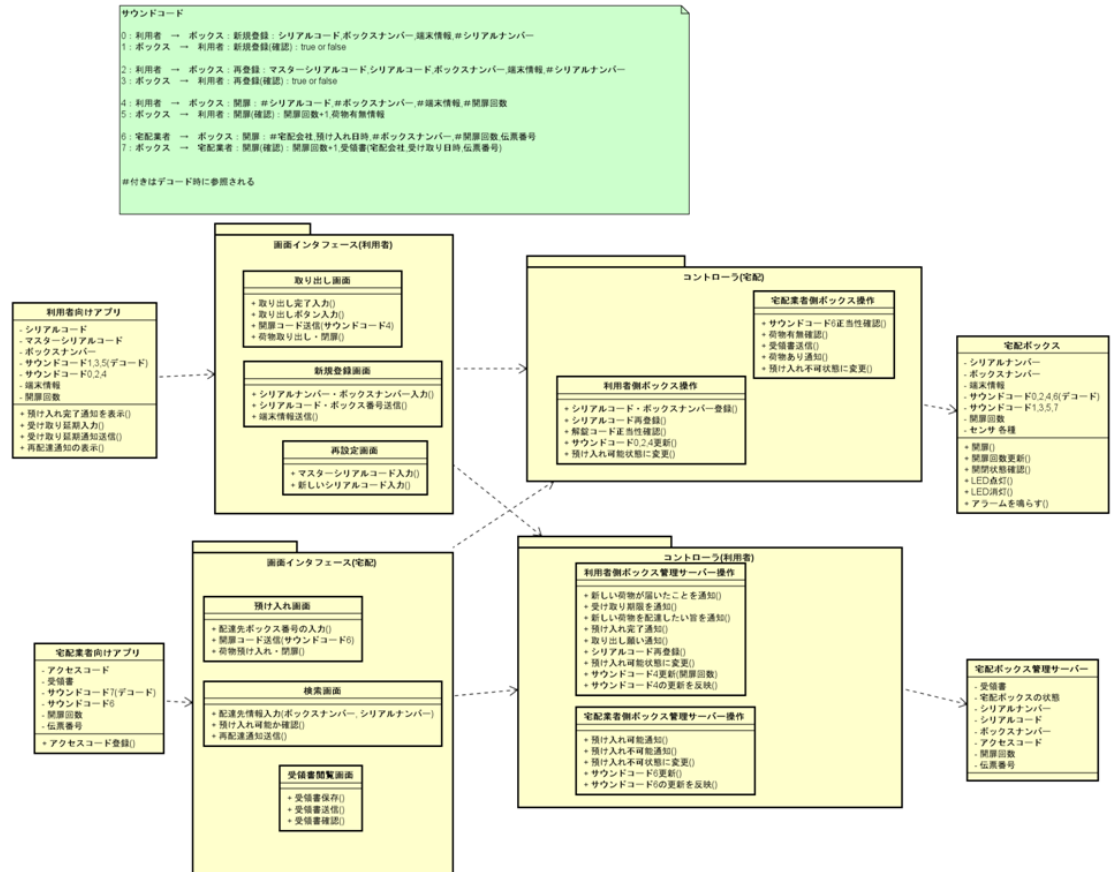


図 5.5. クラス図

クラス図では、シーケンス図の動的なふるまいとは逆に、静的な構造を表現する。各開発対象の持つ属性や操作をクラスとして表現している。また、スマートフォンアプリでは機能画面一つ一つのをクラスとして記述しクラスには画面の属性を記述している。それらを画面インターフェースというパッケージでまとめる。また開発対象同士のつながりを破線の矢印で表現している。クラス図は基本設計と詳細設計として利用している。

優先度の高い機能を抽出したクラス図を図 5.6 に示す。

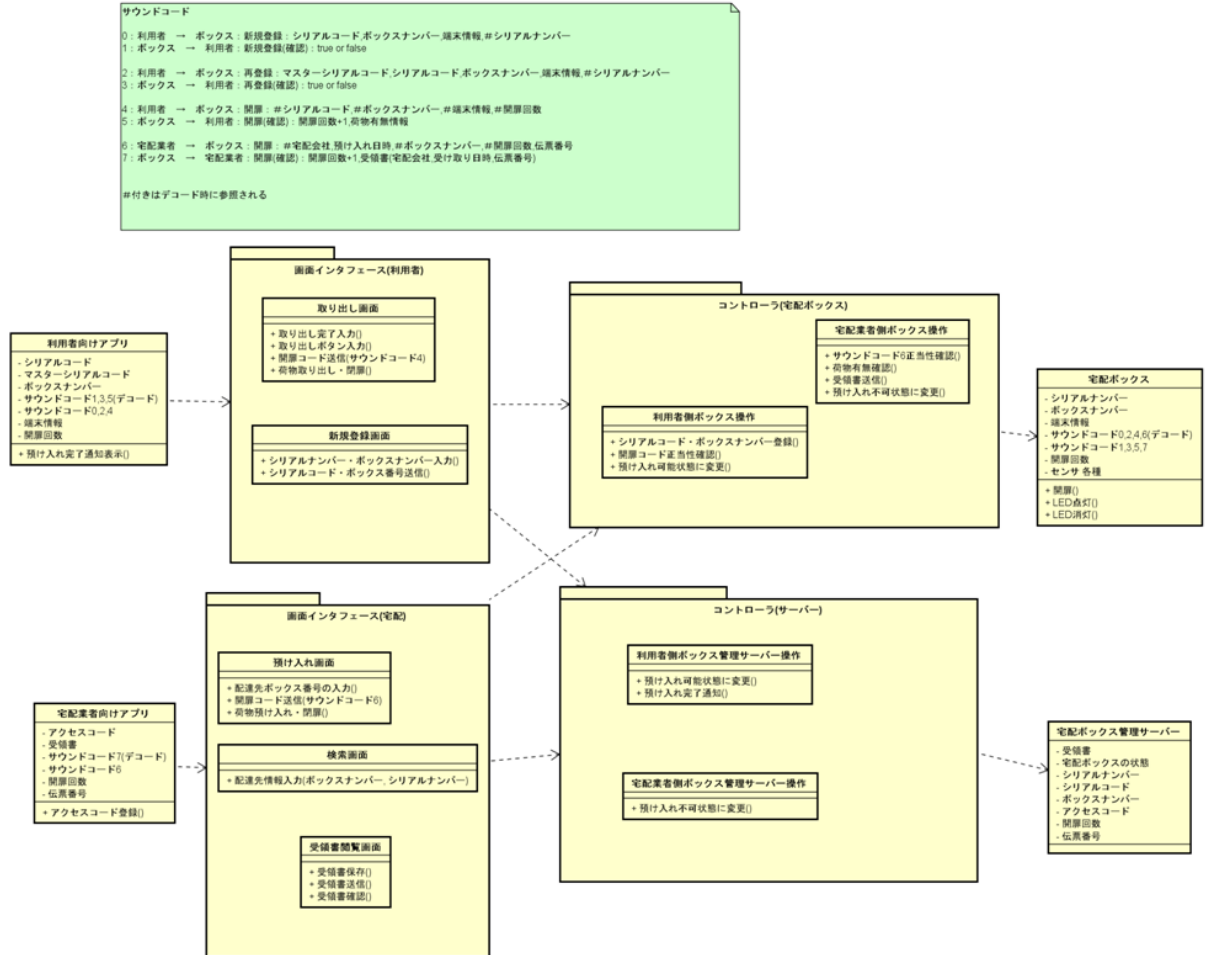


図 5.6. 優先度高いクラス図

実装を容易にするために、クラス図の優先度の高い機能を抽出したクラス図を作成した。優先度の高い機能は、ユースケース図での要件定義を満たすことを条件として抽出した。例えば、シーケンス図で録音対策として記述した機能などはユースケース図には記載されていないため優先度の低い機能として優先度高いクラス図に記載しないこととした。

また、このクラス図を基に、役割分担を行った。役割分担の図を図 5.7 に示す。

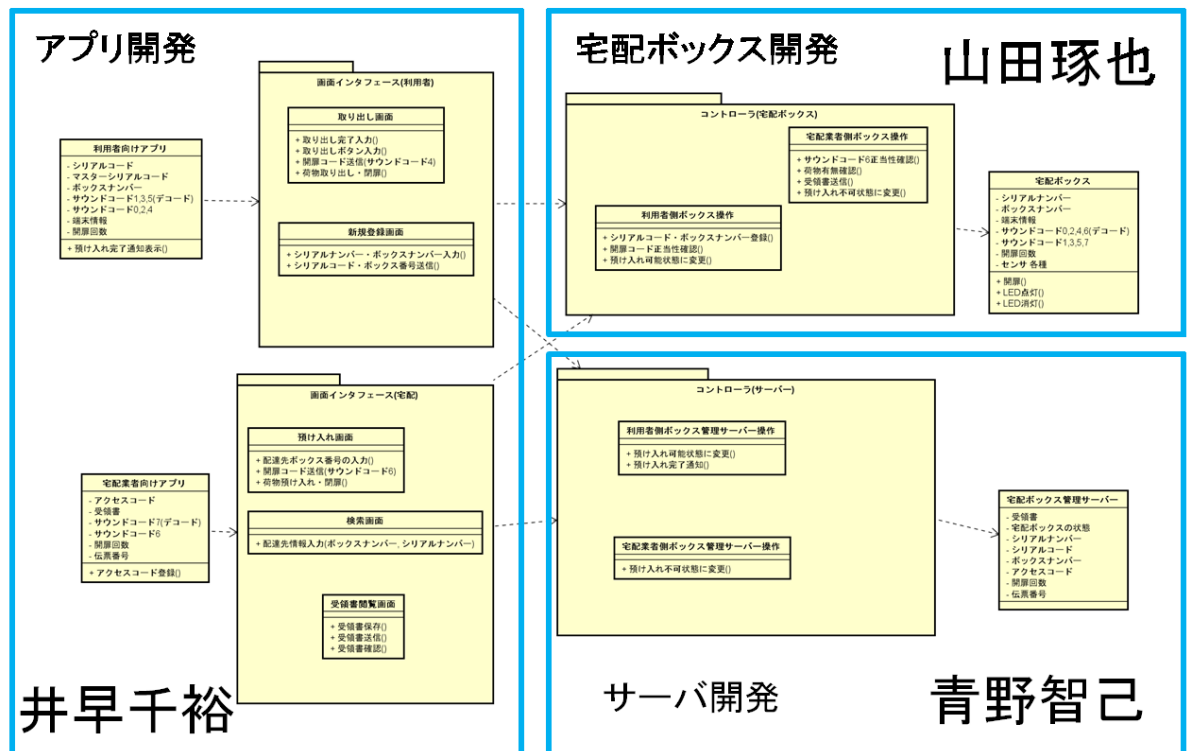


図 5.7. 役割分担

5.2.2. ソフトウェア開発

開発環境 以下に開発環境を示す.

使用機器	Raspberry Pi3 Model B
OS	raspbian9.1
使用言語	php7.0
周辺機器	マイク, スピーカー, マウス, キーボード

開発方法 以下に開発方法を示す.

1. サウンドコードの型の定義を表 5.1 のようにする.
2. 優先度の高いクラス図を確認しながら, 画面インターフェースを作成する.
3. シーケンス図を確認しながら, 預け入れ・取り出しの流れに沿って画面遷移を作成する.
4. 優先度の高いクラス図やサウンドコードの型定義を見て, エンコードするサウンドコードとデコードするサウンドコードを確認しながら, php 上でのエンコード・デコード機能を実装する.

表 5.1. サウンドコードの型

サウンド コード名	利用目的	ライセンスキー
サウンド コード 0	新規登録(利用者 ⇒ボックス)	0x0 シリアルナンバー (7 桁),0x シリアルコード (8 桁),0x00000000,0x ボックス番号 (3 桁)00000
サウンド コード 1	新規登録確認結果(ボックス⇒利用者)	0x1 シリアルコードが正しければ 1 正しくなければ 0(1 桁)000000,0x00000000,0x00000000,0x00000000
サウンド コード 2	再登録(利用者⇒ボックス)	0x2 シリアルナンバー (7 桁),0x シリアルコード (8 桁),0x00000000,0x マスターシリアルコード (5 桁) ボックス番号 (3 桁)
サウンド コード 3	再登録確認結果(ボックス⇒利用者)	0x3true な ら 1false な ら 0(1 桁)000000,0x00000000,0x00000000,0x00000000
サウンド コード 4	開扉(利用者⇒ボックス)	0x4 ボックス番号 (3 桁)0000,0x シリアルコード (8 桁),0x00000000,0x00000000
サウンド コード 5	開扉確認結果(ボックス⇒利用者)	0x5 荷物ありなら 1 荷物なしなら 0(1 桁)000000,0x00000000,0x00000000,0x00000000
サウンド コード 6	開扉(宅配業者⇒ボックス)	0x6 ボックス番号 (3 桁)0000,0x アクセスコード (8 桁),0x00000000,0x00000000
サウンド コード 7	開扉確認結果(ボックス⇒宅配業者)	0x7 荷物ありなら 1 荷物なしなら 0(1 桁)000000,0x00000000,0x00000000,0x00000000

開発結果 以下に開発結果を示す.

利用者向けアプリの新規登録画面を図 5.8 に示す.

図 5.8 の説明を述べる. まず利用者向けアプリを起動し「登録」ボタンを押す. 次にボックス番号・シリアルコード・シリアルナンバーを入力し「確認」ボタンを押すとサウンドコード 0 のライセンスキーが生成され表示される. その後「ボックスに送る」ボタンを押すと php で登録情報が送られると同時に宅配ボックスから流されるサウンドコード 1 の解析が行われ解析情報が表示される. 解析した結果サウンドコード 1 の返答が true であった場合, 「サーバーに送る」ボタンが表示され, false であった場合「戻る」ボタンが表示される. 「サーバーに送る」ボタンを押すと, 解析した情報がサーバに送られ, 送られた情報が表示される. 「戻る」ボタンを押すと一番最初の画面に戻る.

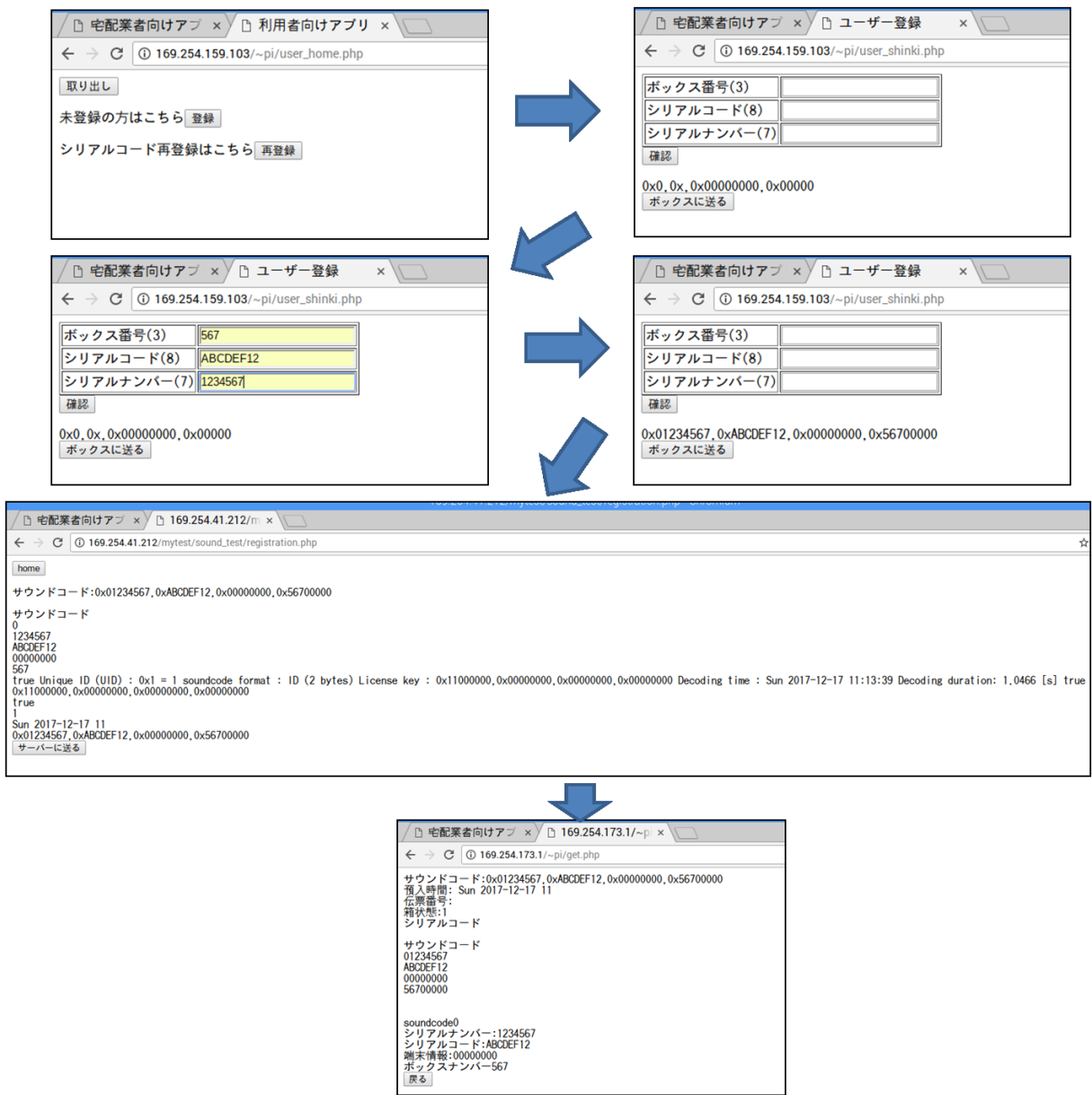


図 5.8. 新規登録画面の遷移図

利用者向けアプリの再登録画面を図 5.9 に示す.

図 5.9 の説明を述べる. まず利用者向けアプリを起動し「登録」ボタンを押す. 次にボックス番号・シリアルコード・シリアルナンバー・マスターシリアルコードを入力し「確認」ボタンを押すとサウンドコード 2 のライセンスキーが生成され表示される. その後「OK?」ボタンを押すと php で登録情報が送られると同時に宅配ボックスから流されるサウンドコード 3 の解析が行われ解析情報が表示される. 解析した結果サウンドコード 1 の返答が true であった場合, 「OK?」ボタンが表示され, false であった場合「戻る」ボタンが表示される. 「OK?」ボタンを押すと, 解析した情報がサーバに送られ, 送られた情報が表示される. 「戻る」ボタンを押すと一番最初の画面に戻る.

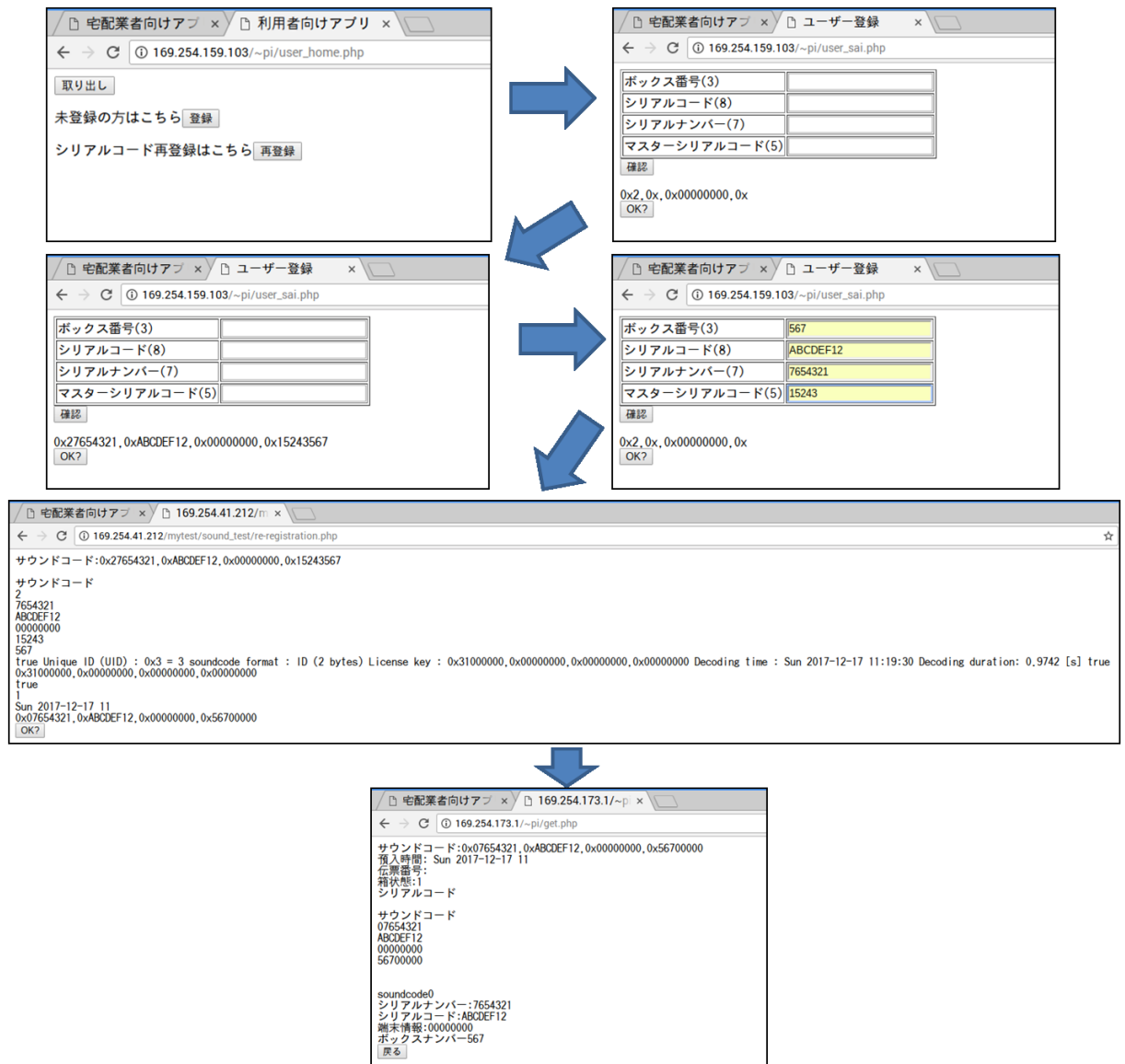


図 5.9. 再登録画面の遷移図

利用者向けアプリの取り出し画面の遷移図を図 5.10 に示す.

図 5.10 の説明を述べる. まず利用者向けアプリを起動し「取り出し」ボタンを押すとサウンドコード 4 が生成されサウンドコード 4 の音声スピーカーから流れる. 次に「データ取得」ボタンを押すと宅配ボックスから流されるサウンドコード 5 の解析が行われ, 解析情報が表示される. 次に「OK?」ボタンを押すと解析情報がサーバに送られ, 送られた情報がアプリ画面に表示される.

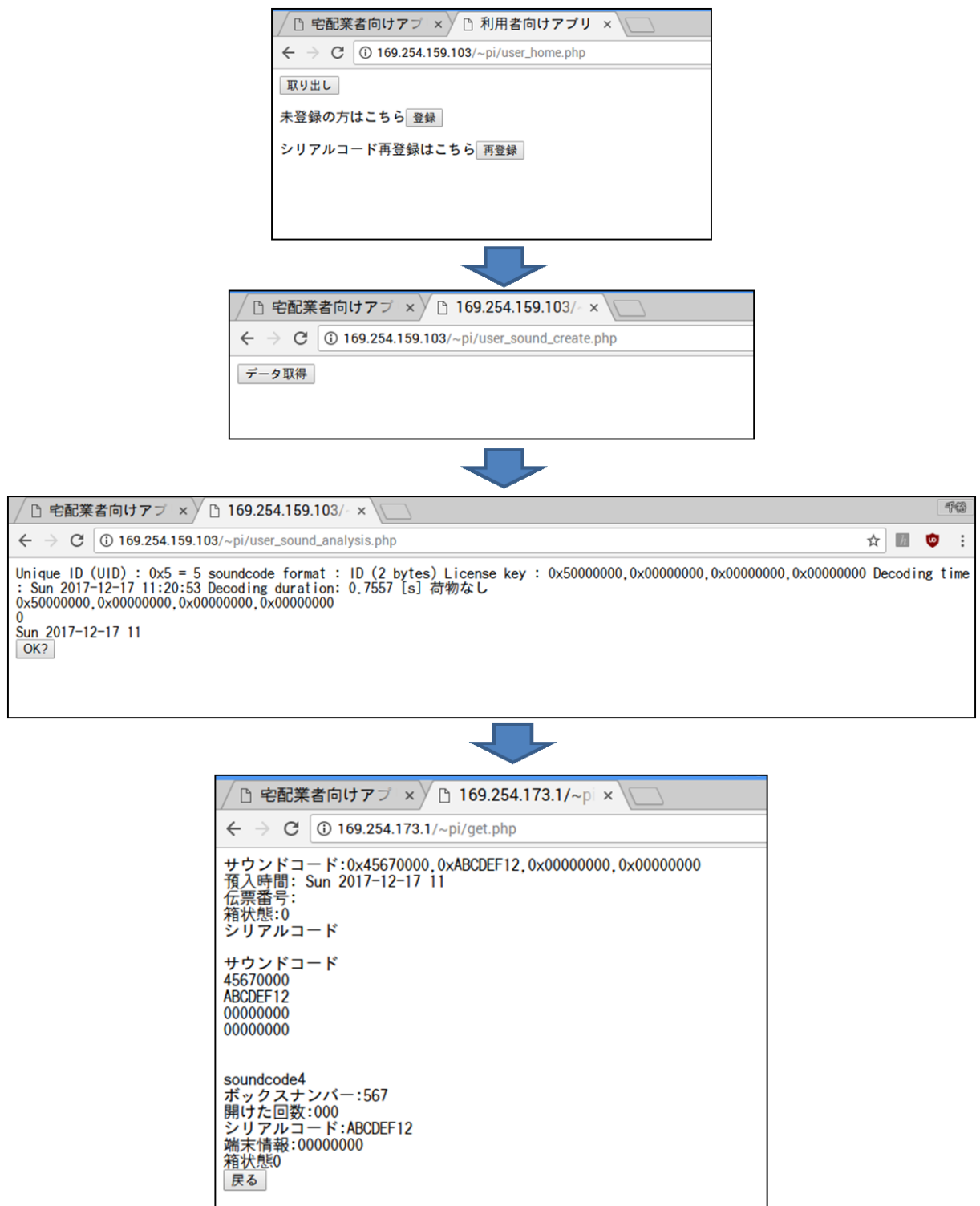


図 5.10. 取り出し画面の遷移図

宅配業者向けアプリの新規登録画面を図 5.11 に示す。

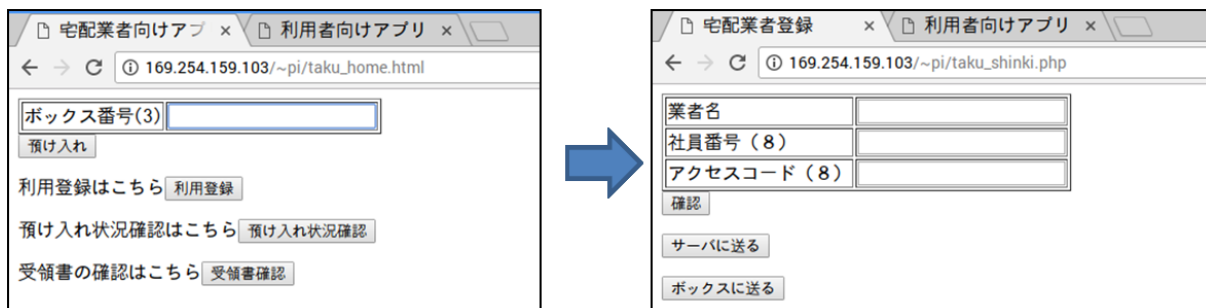


図 5.11. 新規登録画面の遷移図

図 5.11 の説明を述べる. まず宅配業者向けアプリを起動し「利用登録」ボタンを押す. 次に業者名・社員番号・アクセスコードを入力し「ボックスに送る」ボタンを押すと宅配ボックスに php で登録情報が送られる.

宅配業者向けアプリの預け入れ画面の遷移図を図 5.12 に示す.

図 5.12 の説明を述べる. まず宅配業者向けアプリを起動しボックス番号を入力した後「預け入れ」ボタンを押すとサウンドコード 6 が生成されサウンドコード 6 の音声スピーカーから流れる. 次に「受領書取得」ボタンを押すと宅配ボックスから流されるサウンドコード 7 の解析が行われ解析情報がアプリ画面に表示される. 次に「OK?」ボタンを押すと解析情報がサーバに送られ, 送られた情報がアプリ画面に表示される.

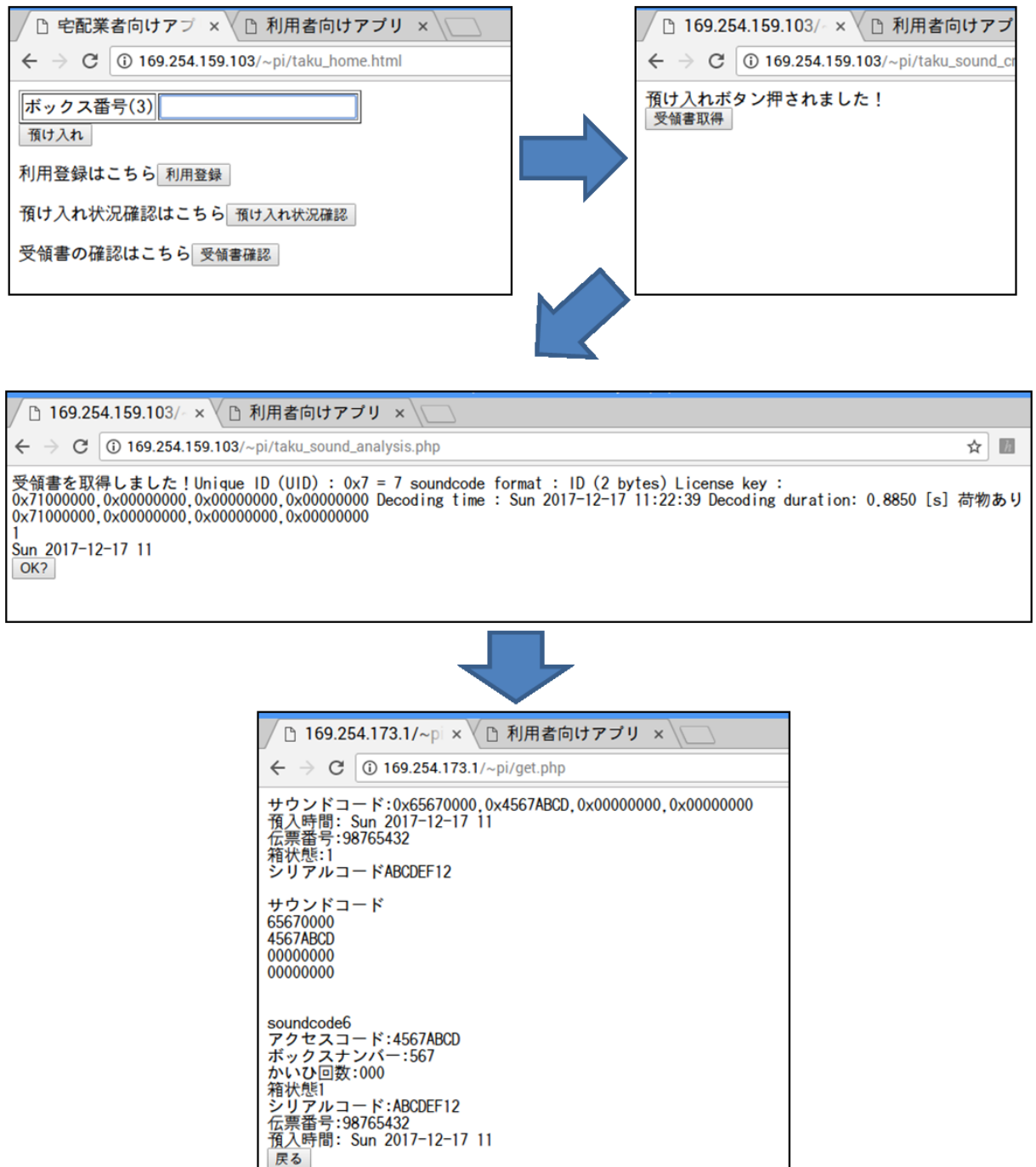


図 5.12. 預け入れ画面の遷移図

宅配業者向けアプリの受領書確認画面を図 5.13 に示す。

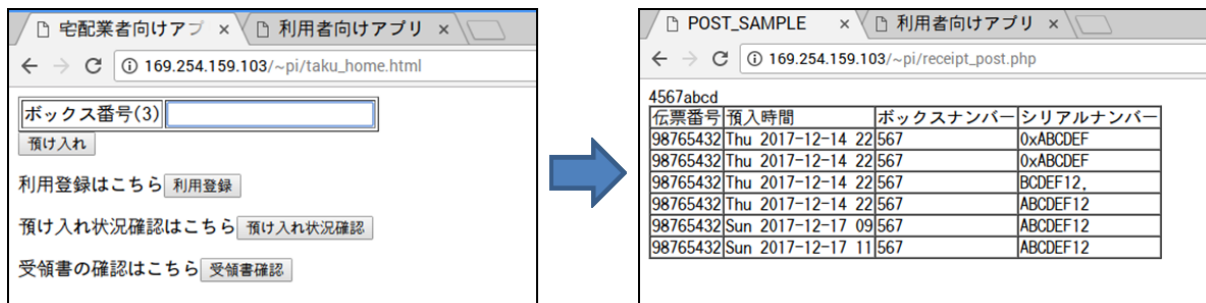


図 5.13. 受領書確認画面

図 5.13 の説明を述べる. 宅配業者向けアプリを起動し「受領書確認」ボタンを押すと, その宅配業者向けアプリ単体の過去の受領書が表示される。

宅配業者向けアプリの預け入れ状況検索画面を図 5.14 に示す。

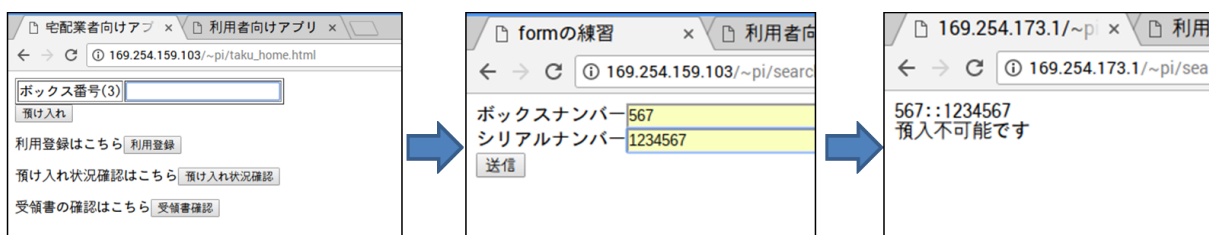


図 5.14. 預け入れ状況検索画面

図 5.14 の説明を述べる. まず宅配業者向けアプリを起動し「預け入れ状況確認」ボタンを押す. 次にボックス番号とシリアルナンバーを入力し「送信」ボタンを押すと, サーバに保管された預け入れ状態が検索され, 「預入可能です」または「預入不可能です」と表示される。

5.2.3. テスト

本研究で行ったテストの結果を示す。

単体テスト

単体テストでは, 機能確認テストを行った. テスト項目とその結果は表 5.2 のとおりである.

結合テスト

結合テストでは, 各開発対象の結合ができていないか確認した. 具体的には, サーバとスマートフォンアプリの通信 (php), 宅配ボックスとスマートフォンアプリの通信 (サウンドコード) がそれぞれできているかを単体テストを行う際に確認した.

表 5.2. 単体テスト結果

機能枠	確認内容	確認結果	確認日	確認者
新規登録	ブラウザ上のテキストフォームからサウンドコード 0 のライセンスキーを作成後 PHP で送信	良好	2017/12/27	井早
	サウンドコード 1 のライセンスキーを取得後, 管理サーバにデータ送信	良好	2017/12/21	井早
再登録	ブラウザ上のテキストフォームからサウンドコード 2 のライセンスキーを作成後 PHP で送信	良好	2017/12/27	井早
	サウンドコード 3 のライセンスキーを取得後, 管理サーバにデータ送信	良好	2017/12/21	井早
預け入れ	ブラウザ上の預け入れ入力によりサウンドコード 6 をエンコードし, 音声出力	良好	2017/12/21	井早
	サウンドコード 7 をデコードできる	良好	2017/12/22	井早
	サウンドコード 7 をデコードした後, 管理サーバにデータ送信	良好	2017/12/22	井早
取り出し	ブラウザ上の取り出し入力によりサウンドコード 4 をエンコードし, 音声出力	良好	2017/12/21	井早
	サウンドコード 5 をデコードできる	良好	2017/12/21	井早
	サウンドコード 5 をデコードした後, 管理サーバにデータ送信	良好	2017/12/21	井早
受領書閲覧	設定したアクセスコードの受領書が見れる	良好	2017/12/21	井早
検索	シリアルコード, ボックスナンバーを PHP で送信後, 箱状態変数が見れる	良好	2017/12/21	井早
新規登録	宅配業者の情報を PHP で送信しサーバに登録する	良好	2017/12/21	井早

総合テスト

総合テストでは, 状態遷移テストを行った. テスト項目とその結果は表 5.3 のとおりである.

また, サウンドコードの音声ファイルをスマートフォンの録音機能 (iPhone6s「ボイスメモ」) で録音し, 録音データでデコードを行ったところ, デコードすることはできなかった.

表 5.3. 総合テスト結果

機能枠	対象	確認内容	確認結果	確認日	確認者
新規登録	利用者アプリ→ボックス	ブラウザテキスト欄のシリアルコード, シリアルナンバー 1, ボックスナンバーからサウンドコード 0 を作成後、ライセンスキーを PHP 送信	良好	2018/1/15	青野, 井早, 山田
	ボックス	取得したライセンスキーを box テーブルのボックスナンバー、シリアルコードと参照して、正当ならテスト項目番号 2.1 へ, そうでないならテスト項目番号 2.2 へ	良好	2018/1/15	青野, 井早, 山田
	ボックス→利用者アプリ	データベースの box テーブルを更新し、サウンドコード 1(true) を音声出力	良好	2018/1/18	青野, 井早, 山田
	ボックス→利用者アプリ	サウンドコード 0(false) を音声出力	良好	2018/1/18	青野, 井早, 山田
	利用者アプリ → 管理サーバ	デコードしたサウンドコード 1 が true ならサウンドコード 0 のライセンスキーを管理サーバへ PHP 送信	良好	2018/1/18	青野, 井早, 山田
	管理サーバ	データベースに box テーブルを作成	良好	2018/1/18	青野, 井早, 山田
	宅配アプリ→ボックス	アクセスコードを PHP で送信後、ボックス側で apli テーブルを作成	良好	2018/1/18	青野, 井早, 山田

機能枠	対象	確認内容	確認結果	確認日	確認者
新規登録	宅配アプリ →管理サーバ	アクセスコードを PHP で送信後、管理サーバ側で apli テーブルを作成	良好	2018/1/18	青野, 井早, 山田
再登録	利用者アプリ →ボックス	ブラウザテキスト欄のシリアルコード, シリアルナンバー 2, ボックスナンバーからサウンドコード 2 を作成後、ライセンスキーを PHP 送信	良好	2018/1/18	青野, 井早, 山田
	ボックス	取得したライセンスキーを box テーブルのボックスナンバー、シリアルコードと参照して、正当ならテスト項目番号 8.1 へ, そうでないならテスト項目番号 8.2 へ	良好	2018/1/18	青野, 井早, 山田
	ボックス→利用者アプリ	データベースの box テーブルを更新、サウンドコード 3(true) を音声出力	良好	2018/1/18	青野, 井早, 山田
	ボックス→利用者アプリ	サウンドコード 3(false) を音声出力	良好	2018/1/18	青野, 井早, 山田
	利用者アプリ →管理サーバ	デコードしたサウンドコード 3 が true ならサウンドコード 2 のライセンスキーを管理サーバへ PHP 送信	良好	2018/1/18	青野, 井早, 山田

機能枠	対象	確認内容	確認結果	確認日	確認者
再登録	管理サーバ	データベースの box テーブルを更新	良好	2018/1/18	青野, 井早, 山田
預け入れ	宅配アプリ →管理サーバ	検索画面からボックスナンバー, シリアルコードを入力して、預け入れ可能か確認	良好	2018/1/18	青野, 井早, 山田
	宅配アプリ →ボックス	ボックスナンバーを入力し、作成したサウンドコード 6 を音声出力	良好	2018/1/18	青野, 井早, 山田
	ボックス	デコードしたサウンドコード 6 が正当でありかつ荷物状態変数が 0 の場合テスト項目番号 13.1 へそうでなければテスト項目番号 13.2 へ	良好	2018/1/18	青野, 井早, 山田
	ボックス→ 宅配アプリ	LED が点灯、スイッチを押すことにより box テーブルの荷物状態変数 1 に変更後サウンドコード 7 を音声出力	良好	2018/1/18	青野, 井早, 山田
	ボックス	LED が点灯しない	良好	2018/1/18	青野, 井早, 山田
	宅配アプリ →管理サーバ	サウンドコード 7 をデコード後、サウンドコード 6 のライセンスキー、荷物状態変数、シリアルナンバー、伝票番号、預け入れ時間を管理サーバに送信	良好	2018/1/18	青野, 井早, 山田
	管理サーバ	receipt テーブルを作成、apli テーブル、box テーブルを更新	良好	2018/1/18	青野, 井早, 山田

機能枠	対象	確認内容	確認結果	確認日	確認者
預け入れ	管理サーバ →利用者アプリ	預け入れ通知	良好	2018/1/18	青野, 井早, 山田
	宅配アプリ →管理サーバ	検索画面からボックスナンバー, シリアルコードを入力して、預け入れ可能か確認	良好	2018/1/18	青野, 井早, 山田
	宅配アプリ →ボックス	サウンドコード 6 を送信し、黄色の LED が点灯しないことを確認	良好	2018/1/18	青野, 井早, 山田
	利用者アプリ →管理サーバ	受領書履歴確認	良好	2018/1/18	青野, 井早, 山田
	利用者アプリ →ボックス	取り出しボタンからサウンドコード 4 を生成して、音声出力	良好	2018/1/18	青野, 井早, 山田
取り出し	ボックス	サウンドコード 4 をデコードして得られたデータと box テーブルの値を参照し、正当ならテスト項目番号 21.1 へ、そうでなければテスト項目番号 21.2 へ	良好	2018/1/18	青野, 井早, 山田
	ボックス→利用者アプリ	LED が点灯、スイッチを押す場合、box テーブルの荷物状態変数 0 に変更後サウンドコード 5 を音声出力スイッチを押さない場合、荷物状態変数 1 のサウンドコード 5 を音声出力	良好	2018/1/18	青野, 井早, 山田

機能枠	対象	確認内容	確認結果	確認日	確認者
取り出し	ボックス	LED が点灯しない	良好	2018/1/18	青野, 井早, 山田
	利用者アプリ → 管理サーバ	サウンドコード 5 をデコード後、サウンドコード 4 のライセンスキー、箱状態変数を管理サーバに PHP 送信	良好	2018/1/18	青野, 井早, 山田
	管理サーバ	box テーブルを更新	良好	2018/1/18	青野, 井早, 山田

5.3 考察

本節では、V 字開発モデルに従ってシステム開発を行う際に作成した UML の考察と、宅配ボックスシステム開発を経て見つかった、サウンドコードを IoT セキュリティシステムに利用する問題点や有効性について述べる。

まず、UML を使用した考察を述べる。UML を用いて設計を行ったことで要求事項の把握や研究メンバーとの認識統一・役割分担をスムーズに行えたと考える。作成した UML はコーディングとテスト両方に利用でき、主にコーディングはクラス図を利用し、テストはシーケンス図を基に作成したテスト項目に従って行われる。また、テストを行った結果要求されている仕様をすべて満たしていることが分かった。

次に問題点について考察する。サウンドコードを宅配ボックスシステムに利用する問題点は、ライセンスキーを指定しないとサウンドコードをデコードできない点である。本研究ではライセンスキーに様々な情報を付加して情報のやり取りをしている。しかし、ライセンスキーを指定しなければならないとなると、事前に送られてくる情報を知っていなければならないということになる。サウンドコード 5 や 7 の荷物有無情報や、サウンドコード 1 や 3 のコードの正当性情報の場合は true(1) または false(0) の二通りであるため、二通りのデコードを実行し、成功しなかった方をタイムアウトさせる方法

をとった。しかし、利用登録や再登録では、送られてくる情報を事前に知り得ないため、サウンドコードが使用できないこととなる。本研究では、宅配ボックスを Raspberry Pi で開発しており、インターネットに接続できるため、php でシリアルコードの再登録機能を実装したが、本来の宅配ボックスシステムでは宅配ボックスをインターネットに接続しないため、ライセンスキーを指定せずにデコードする方法または、サウンドコード通信の通信プロトコルを開発する必要があると考える。

次に、サウンドコードをセキュリティシステムに使用する有効性について考察する。ここではサウンドコードが録音された場合のことについて考察する。サウンドコードは人に聞こえない音であるためデコードされない限り情報が洩れることはない。しかし、情報が目的ではなく、開扉が目的であった場合、開扉用の音声ファイルを録音されるとその音声データで開扉されてしまう可能性があると考えた。そのため、開扉コードであるサウンドコード 4 のライセンスキーを一開扉ごとに変える必要があると考えた。本研究の設計段階で考えた対策は、開扉回数をライセンスキーに付加するというものである。具体的には宅配ボックス側と、サーバ側でそれぞれ開扉回数を保持しておき、サウンドコード 4 のエンコードの際にそのデータをライセンスキーに付加するものである。宅配ボックスとサーバ両方で開扉回数を保持する理由は、開扉回数の相互性を保ち、ライセンスキーを指定したデコードを可能にしかつ、開扉コードの正当性を確認するためである。このように開扉回数を付加し、サウンドコード 4 をエンコードすることで開扉するごとに 1 つ 1 つ違う音声ファイルとなり、録音されても次の開扉ではその録音データは使用できないこととなるため、録音対策に効果があると考えられる。しかし、設計段階で危惧していたこの問題は、実際にシステム開発を終えテストの段階で、録音したファイルではデコードできないことが確認された。そのため、開扉回数の付加は必要ないことがわかった。

このことから、サウンドコードをセキュリティシステムに使用するには、通信プロトコルの開発などの課題はあるが、セキュリティに関しては高い安全性があることが分かった。

第 6 章

あとがき

本研究は, サウンドコード技術を用いた IoT 宅配ボックスシステムを開発することを目的に, システム開発プロセスの V 字開発モデルに従って, 3 人の研究メンバーと宅配ボックスシステム開発を行った. 設計では, UML のうちユースケース図・シーケンス図・クラス図を作成し, システムの要求事項の分析や基本設計・詳細設計を行い, 研究メンバー間での認識統一や役割分担を行うことができた. 開発では, クラス図に従って役割分担をしたことでスムーズに開発を行えた. テストでは, 設計段階で作成したシーケンス図の流れに沿ってテスト項目を作成し検証を行った. その結果要求されている仕様を満たしていることが確認された. また, サウンドコードを IoT セキュリティシステムに使用する問題点と有効性について考察したサウンドコードを IoT セキュリティシステムに使用する問題点はデコードの際デコードするサウンドコードのライセンスキーを指定しなければならない点である. そのため今後の課題はサウンドコード技術のプロトコルの整備であると考え. セキュリティシステムに使用する有効性は, 人に聞こえない音であるため鍵が漏れることがなく, サウンドコードを録音して流しても宅配ボックスの開扉が行われなかったことから, 有効性が高いことがわかった.

謝辞

本研究を進めるにあたり, 懇篤な御指導, 御鞭撻を賜りました本学高橋寛教授に深く御礼申し上げます.

本論文の作成に関し, 詳細なるご検討, 貴重な御教示を頂きました本学樋上喜信准教授に深く御礼申し上げます.

また, 審査頂いた本学岡野大准教授ならびに宇戸寿幸准教授に深く御礼申し上げます.

最後に, 多大な御協力と貴重な御助言を頂いた本学工学部情報工学科情報システム工学講座高橋研究室の諸氏に厚く御礼申し上げます.

参考文献

- [1] 国土交通省, ”宅配便の再配達削減に向けて”, http://www.mlit.go.jp/seisakutokatsu/freight/re_delivery_reduce.html, (参照:平成 30 年 1 月 22 日)

- [2] 内閣府政府広報室, ”「再配達問題に関する世論調査」の概要,” 平成 29 年 12 月, Online, Available, <https://survey.gov-online.go.jp/tokubetu/h29/h29-saihaitatsu.pdf>, (参照:平成 30 年 1 月 22 日)

- [3] 株式会社オーグス総研, 『かんたん UML 増補改訂版』, 加藤正和監修, 株式会社翔泳社, 平成 19 年 6 月 5 日

- [4] FIELDSYSTEM, ”サウンドコード事業”, <http://fieldsystem.co.jp/pc/business/soundcode.php>, (参照:平成 30 年 1 月 22 日)

- [5] webrage, ”VR ブログ「V 字モデルとは？」”, https://webrage.jp/techblog/v_shaped_mode/, (参照:平成 30 年 2 月 7 日)

- [6] 阪田史郎・高田広章編著, 『組込みシステム』, 佐藤政次発行, 株式会社オーム社, 平成 18 年 10 月 15 日

- [7] IT 専 科,”シー ケ ン
ス図”,<http://www.itsenka.com/contents/development/uml/sequence.html>, (参照:
平成 30 年 2 月 7 日)