

# 別居親族向け安否確認システム ミミック

## 内部設計書

第 2. 0 版

株式会社 skiny

平成 29 年 12 月 18 日

## 目 次

<b>1</b>	<b>システム概要</b>	<b>1</b>
<b>2</b>	<b>開発環境</b>	<b>1</b>
<b>3</b>	<b>動作環境</b>	<b>1</b>
<b>4</b>	<b>モジュール構成</b>	<b>1</b>
<b>5</b>	<b>モジュール仕様</b>	<b>3</b>
5.1	loginPage.html . . . . .	3
5.2	userMyPage.html . . . . .	3
5.3	history.html . . . . .	4
5.4	contractSetting.html . . . . .	4
5.5	useManagement.html . . . . .	5
5.6	questionAndAnswer.html . . . . .	5
5.7	contractInformation.html . . . . .	5
5.8	adminMyPage.html . . . . .	6
5.9	registerContract.html . . . . .	6
5.10	searchContract.html . . . . .	7
5.11	resultSearchContract.html . . . . .	7
5.12	updateContract.html . . . . .	7
5.13	login.php . . . . .	8
5.14	history.php . . . . .	9
5.15	getConf.php . . . . .	9
5.16	setConf.php . . . . .	10
5.17	reflectionBoxConf.php . . . . .	10
5.18	registerContract.php . . . . .	11
5.19	searchContract.php . . . . .	11
5.20	updateContract.php . . . . .	12
5.21	registerHistroy.php . . . . .	13
5.22	openSensor.py . . . . .	13
5.23	dataReflection.py . . . . .	13
5.24	dataTransmisson.py . . . . .	14
5.25	alarm.py . . . . .	14
5.26	leaveSensor.py . . . . .	14
<b>6</b>	<b>データベース設計</b>	<b>14</b>
6.1	contracts . . . . .	14
6.2	users . . . . .	15
6.3	configurations . . . . .	15
6.4	histories . . . . .	15
6.5	administrators . . . . .	15

## 1 システム概要

本システムは、別居している高齢者を親族に持つ方に向けた安否確認システムである。システムは Web システムであり、高齢者の自宅に設定した薬箱の使用状況をこのシステムでモニタリングできるようにすることで実現する。

## 2 開発環境

本システムの開発環境を以下に示す。

使用言語	HTML、CSS、PHP、Python
DBMS	MySQL14.14
サーバソフトウェア	Apache HTTP Server 2.4.29
バージョン管理	GitHub

## 3 動作環境

本システムの動作環境を以下に示す。

- 保証ブラウザ

GoogleChrome 63.0

## 4 モジュール構成

本システムのモジュール構成を図 1, 図 2 に示す。本設計書では、ファイル単位でモジュールを規定し、矢印に対して始点のモジュールから終点のモジュールの呼び出しを行う。

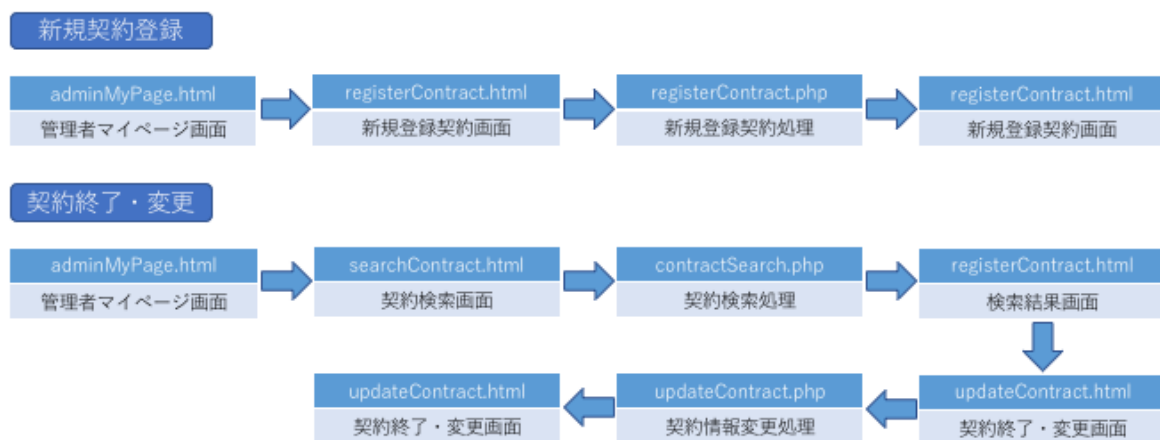


図 1: モジュール構成 1

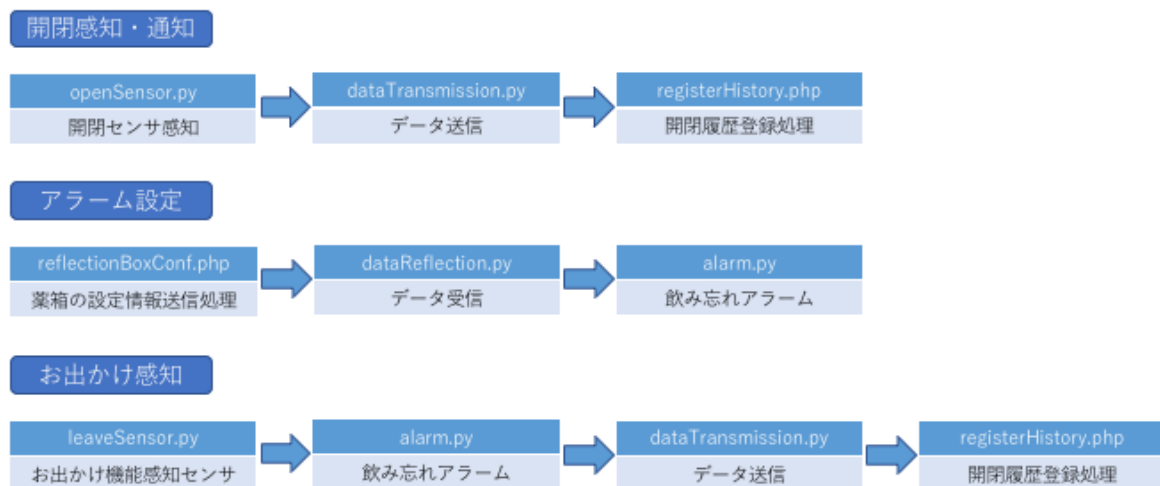


図 2: モジュール構成 2

## 5 モジュール仕様

各モジュールの仕様を以下に示す.

「」は画面のボタン名, “ ” は変数名, [ ] はデータベースの属性名を表す.

### 5.1 loginPage.html

【名称】

ログイン画面

【概要】

契約者はここから個人のページへとログインすることができる.

【入出力】

テキストボックスに契約者 ID, パスワードを入力する. その上で, 「ログイン」を選択する.

【処理フロー】

- 「ログイン」をクリックすると, login.php が呼び出される.
- TRUE 判定が返されると, login.php で処理し, userMyPage.html を呼び出す.
- FALSE 判定が返されると, エラー表示がされます. ただし再度, 入力することができる.

### 5.2 userMyPage.html

【名称】

契約者マイページ

【概要】

契約者のマイページが表示される.

【入出力】

契約者は「履歴」「使用設定」「使い方」「Q & A」「契約情報」を選択できる. また, 「ログアウト」を選択できる.

【処理フロー】

- 「履歴」をクリックすると, history.html を呼び出す.
- 「使用設定」をクリックすると, contractSetting.html を呼び出す.
- 「使い方」をクリックすると, useManagement.html を呼び出す.
- 「Q & A」をクリックすると, questionAndAnswer.html を呼び出す.
- 「契約情報」をクリックすると, contractInformation.html を呼び出す.
- 「ログアウト」をクリックすると, loginPage.html を呼び出す.

### 5.3 history.html

【名称】

履歴閲覧画面

【概要】

契約者は薬箱の使用状況を閲覧することができる。最初の画面では、履歴閲覧日から 1 週間の履歴が表示される。

【入出力】

以下の操作を行うことができる。

- カレンダーの左右にある「矢印」ボタンを押すことで閲覧する情報を履歴閲覧日から 1 週間間隔で変更することができる。
- カレンダーの日付を押すことで、その日の詳細な情報を閲覧することができる。
- 画面左上にある「戻る」ボタンを押すことで「契約者マイページ」へ戻ることができる。

【処理フロー】

- カレンダーの日付をクリックすると、boxHistory.php を呼び出す。
- 「戻る」ボタンをクリックすると、userMyPage.html を呼び出す。

### 5.4 contractSetting.html

【名称】

使用設定画面

【概要】

契約者は薬箱のアラームの時間や通知についての設定を行うことができる。

【入出力】

アラーム設定では以下の操作を行う。

- ドロップダウンボタンを押すことで時間を変更することができる。
- チェックボックスによってアラームの on / off を変更することができる。

通知設定では以下の操作を行う。

- E メール、LINE でそれぞれ通知を受け取るかどうかをトグルボタンを押すことで変更することができる。
- チェックボックスによって通知の受け取り方法を変更することができる。
- 通知を受け取る E メール、LINE がテキストボックスに表示し、テキスト入力によって変更することができる。

【処理フロー】

- それぞれの設定を行った後、「変更」をクリックできる。
- 「変更」をクリックすると、setConf.php を呼び出す。
- 「戻る」ボタンをクリックすると、userMyPage.html を呼び出す。

## 5.5 useManagement.html

【名称】

「使い方」閲覧画面

【概要】

契約者はこのサービスの基本的な利用方法を閲覧することができる。

【処理フロー】

- 「戻る」ボタンをクリックすると, userMyPage.html を呼び出す。

## 5.6 questionAndAnswer.html

【名称】

「Q & A」閲覧画面

【概要】

契約者は利用する上での基本的な質問とその質問に対する回答を閲覧することができる。

【処理フロー】

- 「戻る」ボタンをクリックすると, userMyPage.html を呼び出す。

## 5.7 contractInformation.html

【名称】

契約情報閲覧及び変更画面

【概要】

契約者は現在利用されている契約情報を閲覧することができる。また, 以下について契約内容の変更を行うことができる。

【入出力】

契約者は以下の契約情報について, テキストによる入力を行い変更を行うことができる。

【契約情報の変更可能項目】

- パスワード
- メールアドレス
- 電話番号

【処理フロー】

- 契約情報の変更可能項目について, テキストボックスに入力後, 「変更」をクリックできる。
- 「変更」をクリックすると, updateContract.php を呼び出す。

## 5.8 adminMyPage.html

### 【名称】

管理者マイページ画面

### 【概要】

管理者マイページが表示される。

### 【入出力】

管理者は「新規契約登録」「契約検索」を選択できる。また「ログアウト」を選択できる。

### 【処理フロー】

- 「新規契約登録」をクリックすると,registerContract.html を呼び出す。
- 「契約検索」をクリックすると,searchContracrt.html を呼び出す。
- 「ログアウト」をクリックすると,loginPage.html を呼び出す。

## 5.9 registerContract.html

### 【名称】

新規契約登録画面

### 【概要】

新規契約登録を行う初期画面が表示される。

### 【入出力】

管理者は各テキストボックスに登録したい契約の情報を入力する。入力する契約の情報を以下に示す。

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者パスワード
- 契約者メールアドレス
- 契約者電話番号
- ラズベリーパイ IP アドレス

### 【処理フロー】

- 各テキストボックスに入力後,「登録」をクリックできる。
- 「登録」をクリックすると registerContract.php を呼び出す。
- TRUE 判定が返されると,登録完了を通知するポップアップを表示し,その後 registerContract.html を呼び出す。
- FALSE 判定が返されると,エラー表示がされる。ただし,再入力はいえる。
- 画面右上の「安否確認システムミミック (システムロゴ)」をクリックすると,adminMyPage.html を呼び出す。



## 5.10 searchContracrt.html

【名称】

契約検索画面

【概要】

契約検索を行う初期画面が表示される。

【入出力】

管理者は契約の情報のキーワードの入力を行うことができる。入力できる契約の情報を以下に示す。

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者メールアドレス
- 契約者電話番号
- ラズベリーパイ IP アドレス

【処理フロー】

- 各テキストボックスに入力後,「検索」をクリックできる。
- 「検索」をクリックすると contractSearch.php で処理し,resultSearchContract.html を呼び出す。
- 画面右上の「安否確認システムミミック (システムロゴ)」をクリックすると,adminMyPage.html を呼び出す。

## 5.11 resultSearchContract.html

【名称】

検索結果画面

【概要】

契約検索の検索内容と合致した契約の契約者 ID, 契約者名, 薬箱利用者名が一覧で表示される。

【処理フロー】

- 契約内容をクリックすると,updateContract.html を呼び出す。
- 画面右上の「安否確認システムミミック (システムロゴ)」をクリックすると,adminMypaget.html を呼び出す。

## 5.12 updateContract.html

【名称】

契約終了・変更画面

【概要】

契約情報の変更, 契約終了を行う画面です.

【入出力】

契約の情報がテキストボックスに表示され, 変更を行うことができる. また, ユーザの契約終了を行うことができる. 変更できる契約の情報を以下に示す.

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者パスワード
- 契約者メールアドレス
- 契約者電話番号
- ラズベリーパイ IP アドレス

【処理フロー】

- 各テキストボックスの情報を变更后, 「変更」をクリックできる.
- 「変更」をクリックすると, updateContract.php を呼び出す.
- TRUE 判定が返されると, 変更完了を通知するポップアップを表示し, その後 updateContract.html を呼び出す.
- FALSE 判定が返されると, エラー表示がされる. ただし, 再入力は行える.
- 「契約終了」をクリックすると, 契約終了の完了を通知するポップアップを表示し, その後 resultSearchContract.html を呼び出す.
- 画面右上の「安否確認システムミミック (システムロゴ)」をクリックすると, adminMy-page.html を呼び出す.

## 5.13 login.php

【名称】

ログイン認証処理

【概要】

パスワードの正否を判定する.

【入出力】

login.html から以下の認証情報が入力される.

- 契約者 ID
- パスワード

パスワードが正しい場合は TRUE, 正しくない場合は FALSE を login.html に出力する.

【処理フロー】

- データベースに対して入力されたユーザ ID と一致する [user\_id] を検索し、パスワードと [password] を照合.
- パスワードが正しい場合、TRUE 判定を返却.
- ユーザが見つからない、またはパスワードが正しくない場合、FALSE 判定を返却.

## 5.14 history.php

【名称】

履歴取得処理

【概要】

データベースから薬箱の開閉時間や状態を取得する.

【入出力】

history.html から以下の契約者情報が入力される.

- 契約者 ID

history.html へ以下の薬箱の履歴情報を出力する.

- 開閉時間
- 状態

【処理フロー】

- 入力された [contracts\_id] を “id” に格納する.
- “id” を基に histories テーブルを検索.
- データベースから [acted\_at],[state] を取得して出力する.

## 5.15 getConfig.php

【名称】

設定情報取得処理

【概要】

データベースに格納された薬箱の設定情報を取得する.

【入出力】

以下の情報が入力される.

- 契約 ID

以下の情報を出力する.

- アラーム設定時刻
- アラーム通知の有無

【処理フロー】

- 入力された [contracts.id] を “id” に格納する.
- “id” を基に configurations テーブルを検索する.
- 検索したユーザの設定情報を出力.

## 5.16 setConf.php

【名称】

設定情報格納処理

【概要】

configuration テーブルに薬箱の設定情報を格納する.

【入出力】

以下の情報が入力される.

- 契約 ID

以下の情報を出力する.

- TRUE または FALSE

【処理フロー】

- 入力された設定情報がデータベースで指定された型に合っているかを TRUE か FALSE で判断.
- TRUE であれば configurations テーブルに設定情報を格納し TRUE を返却.
- FALSE であれば何もせずに FALSE を返却.

## 5.17 reflectionBoxConf.php

【名称】

薬箱の設定情報送信処理

【概要】

薬箱にアラームの時刻を送信する.

【入出力】

以下の情報が入力される.

- 契約 ID

以下の情報を出力する.

- アラーム時刻

【処理フロー】

- 入力された [contracts.id] を “id” に格納する.
- “id” を基に configurations テーブルを検索する.
- 薬箱に [morning\_time],[noon\_time],[evening\_time],[night\_time] を送信.

## 5.18 registerContract.php

【名称】

新規契約登録処理

【概要】

新規契約情報をデータベースに登録する。

【入出力】

以下の契約情報が入力される。

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者パスワード
- 契約者メールアドレス
- 契約者電話番号
- ラズベリーパイ IP アドレス

以下の情報を出力する。

- TRUE または FALSE

【処理フロー】

- 入力された契約情報を “contracts” に格納する,
- “contracts” を contracts テーブル, users テーブルに登録する。

## 5.19 searchContract.php

【名称】

契約検索処理

【概要】

データベースから契約情報を取得する。

【入出力】

以下の契約情報が入力される。

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者メールアドレス
- 契約者電話番号

- ラズベリーパイ IP アドレス

以下の情報を出力する。

- 契約者 ID
- 契約者名
- 薬箱利用者名

【処理フロー】

- 入力された契約情報を “contracts” に格納する。
- “contracts” を用いて contracts テーブルと users テーブルから契約情報を検索する。
- 検索した契約情報から [user\_id], [name], [partner] を出力する。

## 5.20 updateContract.php

【名称】

契約情報変更処理

【概要】

変更された契約情報をデータベースに登録する。

【入出力】

以下の情報が入力される。

- 契約者 ID
- 契約者名
- 薬箱利用者名
- 契約者パスワード
- 契約者メールアドレス
- 契約者電話番号
- ラズベリーパイ IP アドレス

以下の情報を出力する。

- TRUE または FALSE

【処理フロー】

- 入力された設定情報を “setting” に格納する。
- “setting” がデータベースで指定された型に合っているかを TRUE か FALSE で判断。
- TRUE であれば、入力された契約情報を引数とし、変更された契約情報を contracts テーブル, users テーブルに登録。
- FALSE であれば何もせずに FALSE を返却。

## 5.21 registerHistroy.php

【名称】

開閉履歴登録処理

【概要】

薬箱から受け取った開閉履歴をデータベースへ登録する.

【入出力】

以下の情報が入力される.

- 開閉履歴

以下の情報を出力する.

- 開閉履歴

【処理フロー】

- 薬箱から送られてきた [contracts\_id] を “id” に格納する.
- “id” で契約を検索.
- 開閉時間と薬箱の状態を histories テーブルの [acted\_at], [state] に登録.

## 5.22 openSensor.py

【名称】

開閉センサ感知

【概要】

薬箱に設置された開閉センサから送られてくる情報を受信する.

【処理フロー】

- センサが作動するとデータ送信モジュールを呼び出し、開閉センサ作動情報”open” を渡す.
- 飲み忘れアラームに”open” を渡す.

## 5.23 dataReflection.py

【名称】

データ受信

【概要】

サーバから送られてきたデータをアラーム時間を変数に格納する.

【処理フロー】

- サーバから送られてきたアラーム時間を変数に格納する.
- 朝に鳴らす時間を変数 “morning” に、昼は “noon”，夜は “afternoon”，就寝前は “night” に格納する.

## 5.24 dataTransmisson.py

【名称】

データ送信

【概要】

センサから送られてきたデータをサーバに送信する。

【処理フロー】

- 開閉センサ感知モジュール，またはお出かけ機能センサ感知モジュールから”open” または”leave” を受け取りそれを現在時刻と共にサーバーへと送信する。
- ”open” を送る際はアラーム設定時間を過ぎてから 30 分後に送信する。

## 5.25 alarm.py

【名称】

飲み忘れアラーム

【概要】

設定された時間になっても薬箱が開かれなかった場合，アラーム音声を流す。

【処理フロー】

- 変数 “morning”，“noon”，“afternoon”，“night” に格納されているそれぞれの時間までに開閉センサ感知モジュールから”open” が渡されていなかった場合，アラーム音声 2 回を 5 分間隔で 4 セット鳴らす。

## 5.26 leaveSensor.py

【名称】

お出かけ機能感知センサ

【概要】

薬箱に設置されたお出かけ機能センサから送られてくる情報を受信する。

【処理フロー】

- センサが作動するとデータ送信モジュールを呼び出し，お出かけセンサ作動情報”leave” を渡す。

# 6 データベース設計

サーバで管理するデータベースの構成を定義する。

## 6.1 contracts

契約情報を格納する。各属性の詳細な定義を表 1 に示す。



表 1: contracts テーブル

属性	データ型	NULL	主キー/外部キー
id	char(8)	no	主キー
user_id	char(8)	no	外部キー:users. id
configuration_id	char(8)	no	外部キー:configurations. id
partner	varchar(64)	yes	
ip_address	int	yes	
started_at	timestamp	yes	
closed_at	timestamp	yes	
created_at	timestamp	no	
updated_at	timestamp	yes	

## 6.2 users

ユーザ情報を格納する. 各属性の詳細な定義を表 2 に示す.

表 2: users テーブル

属性	データ型	NULL	主キー/外部キー
id	char(8)	no	主キー
name	varchar(64)	yes	
password	varchar(16)	yes	
mail	varchar(64)	no	
phone_number	int	yes	
address	varchar(64)	yes	
registered_at	timestamp	yes	
deleted_at	timestamp	yes	
created_at	timestamp	no	
updated_at	timestamp	yes	

## 6.3 configurations

設定情報を格納する. 各属性の詳細な定義を表 3 に示す.

## 6.4 histories

薬箱の使用履歴情報を格納する. 各属性の詳細な定義を表 4 に示す.

## 6.5 administrators

管理者情報を格納する. 各属性の詳細な定義を表 5 に示す.

表 3: configurations テーブル

属性	データ型	NULL	主キー/外部キー
id	char(8)	no	主キー
contract_id	char(8)	no	外部キー:contracts. id
morning_time	time	yes	
noon_time	time	yes	
evening_time	time	yes	
night_time	time	yes	
mail	varchar(64)	yes	
mail_after_blank_time	enum('on','off')	yes	
blank_time_for_mail	time	yes	
mail_once_a_day	enum('on','off')	yes	
line	varchar(64)	yes	
line_after_blank_time	enum('on','off')	yes	
blank_time_for_line	time	yes	
line_once_a_day	enum('on','off')	yes	
created_at	timestamp	no	
updated_at	timestamp	yes	

表 4: histories テーブル

属性	データ型	NULL	主キー/外部キー
contract_id	char(8)	no	外部キー:contracts. id
acted_at	timestamp	no	
state	enum('opened','no_response', 'go_out','return_home')	no	

表 5: administrators テーブル

属性	データ型	NULL	主キー/外部キー
id	char(8)	no	主キー
name	varchar(64)	no	
password	varchar(16)	no	
created_at	timestamp	no	
updated_at	timestamp	yes	