

別居親族向け安否確認システム ミミック

内部設計書

株式会社 skiny

平成 29 年 12 月 11 日

目 次

1	システム概要	1
2	開発環境	1
3	動作環境	1
4	モジュール仕様	1
4.1	loginPage.html	1
4.2	userMyPage.html	2
4.3	history.html	2
4.4	registerSetting.html	3
4.5	useManagement.html	3
4.6	questionAndAnswer.html	3
4.7	registerInformation.html	4
4.8	adminMyPage.html	4
4.9	registerContract.html	4
4.10	searchContract.html	5
4.11	resultSearchContract.html	6
4.12	updateContract.html	6
4.13	login.php	7
4.14	searchUser.php	7
4.15	history.php	7
4.16	getConf.php	8
4.17	setConf.php	8
4.18	reflectionBoxConf.php	8
4.19	registerContract.php	8
4.20	contractSearch.php	9
4.21	updateContract.php	9
4.22	registerHistory.php	9
4.23	openSensor.py	9
4.24	dataReflection.py	10
4.25	dataTransmission.py	10
4.26	boxAlarm.py	10
4.27	leaveSensor.py	10
5	データベース設計	11
5.1	contracts	11
5.2	users	11
5.3	configurations	11
5.4	histories	12
5.5	administrators	12

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 モジュール仕様

4.1 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ます。

【入出力】

テキストボックスに契約者 ID, パスワードを入力します。その上で、「ログイン」を選択します。

【処理フロー】

- 「ログイン」をクリックすると、.php が呼び出されます。
- TRUE 判定が返されると、.php で処理し、userMyPage.html を呼び出します。
- 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ます。

4.2 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ます。

【入出力】

契約者は「履歴」「使用設定」「使い方」「Q & A」「契約情報」を選択できます。また、「ログアウト」を選択できます。

【処理フロー】

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

4.3 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ます。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ます。

【入出力】

以下の操作を行うことができます。

- 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ます。
- 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ます。
- 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」へ戻ることができます。

【処理フロー】

- カレンダーの日付をクリックすると, boxHistory.php が呼び出されます。
- 「戻る」ボタンをクリックすると, userMyPage.html を呼び出します。

4.4 registerSetting.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ます。

【入出力】

アラーム設定では以下の操作を行います。

- 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ます。
- 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ます。

通知設定では以下の操作を行います。

- 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ます。
- 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ます。
- 通知を受け取る E メール, LINE がテキストボックスに表示されており, テキスト入力によって変更することができます。

【処理フロー】

- それぞれの設定を行った後, 「変更」をクリックできます。
- 「変更」をクリックすると, setConf.php が呼び出されます。
- 「戻る」ボタンをクリックすると, userMyPage.html を呼び出します。

4.5 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ます。

【処理フロー】

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

4.6 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ます。

【処理フロー】

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

4.7 registerInformation.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ます。また、以下について契約内容の変更を行うことができます。

【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ます。

【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」をクリックできます。
- 「変更」をクリックすると、updateContract.php が呼び出されます。

4.8 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 を呼び出される。
- 「契約検索」をクリックすると、searchContract.html を呼び出される。
- 「ログアウト」をクリックすると、loginPage.html を呼び出される。

4.9 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 が呼び出される.
- 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 を呼び出される.
- 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.
- 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 を呼び出される.

4.10 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 が呼び出される。
- 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 を呼び出される。

4.11 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 が呼び出される。
- 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,adminMypaget.html を呼び出される。

4.12 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 が呼び出される。

- 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.

4.13 login.php

【名称】

ログイン認証処理

【概要】

入力したパスワードの正否を判定する.

【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.

4.14 searchUser.php

【名称】

契約情報確認処理

【概要】

データベースから契約情報を取得する.

【処理フロー】

- ・ contracts テーブルと users テーブルから契約情報を検索し出力.

4.15 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.

【処理フロー】

- ・ データベースから [acted_at],[state] を取得して出力.

4.16 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。

【処理フロー】

- ・ configurations テーブルからログインしているユーザの [id] を検索。
- ・ 検索したユーザの設定情報を取得。

4.17 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。

【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。

4.18 reflectionBoxConf.php

【名称】

薬箱の設定情報送信処理

【概要】

薬箱にアラームの設定時刻を送信する。

【処理フロー】

- ・ 薬箱に [morning_time],[noon_time],[evening_time],[night_time] を送信。

4.19 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。

【処理フロー】

- ・ 入力された契約情報を引数とし、新規契約情報を contracts テーブル, users テーブルに登録。

4.20 contractSearch.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.

【処理フロー】

- ・ contracts テーブルと users テーブルから契約情報を検索し, [user_id], [name], [partner] を出力.

4.21 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.

【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, 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.

4.22 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.

【処理フロー】

- ・ [contract_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.

4.23 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” を渡す.
- ・ 飲み忘れアラームモジュールに” 開閉センサ作動情報” を渡す.

4.24 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。

4.25 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”センサ動作情報”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。

4.26 boxAlarm.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”開閉センサ作動情報”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3セット鳴らす。

4.27 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”を渡す。

5 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.

5.1 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	

5.2 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	

5.3 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.

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	

5.4 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.

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	

5.5 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 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	