出退勤管理システム

water drop

小西　孝海

2018

内容

[1. はじめに 2](#_Toc529090483)

[2. 目的 2](#_Toc529090484)

[3. 用語の定義 2](#_Toc529090485)

[4. システム構成 3](#_Toc529090486)

[5. デプロイ 3](#_Toc529090487)

[6. URL設計 4](#_Toc529090488)

[7. アプリ概要 5](#_Toc529090489)

[8.1. 出勤 5](#_Toc529090498)

[8.2. 設定 6](#_Toc529090499)

[8. データ設計 7](#_Toc529090500)

[9.1. タイムカードテーブル 7](#_Toc529090502)

[9. 機能概要 7](#_Toc529090503)

[10.1. 認証機能 8](#_Toc529090505)

[10.2. 権限管理機能 8](#_Toc529090506)

[10.3. ログイン機能 10](#_Toc529090507)

[10.4. 出退勤データ管理機能 10](#_Toc529090508)

[9.3.1. 社員データ管理機能 11](#_Toc529090512)

[9.3.2. タイムカード機能 11](#_Toc529090513)

[9.3.3. 有休管理機能 12](#_Toc529090514)

[9.3.4. 作業内容管理機能 12](#_Toc529090515)

[9.3.5. 給与明細作成機能 12](#_Toc529090516)

[10.5. 交通費管理機能 12](#_Toc529090517)

[10.6. システム管理機能（ここのボリュームがやばいことになりそう） 12](#_Toc529090518)

[9.5.1. カレンダー作成機能 13](#_Toc529090521)

[9.5.2. インポート機能 13](#_Toc529090522)

[9.5.3. エクスポート機能 13](#_Toc529090523)

[9.5.4. バックアップ機能 13](#_Toc529090524)

[10.7. RPA抑止機能 13](#_Toc529090525)

[10. 付録 13](#_Toc529090526)

[13.1. djangoでメールを送信する 13](#_Toc529090530)

[11. メモ 14](#_Toc529090531)

# はじめに

本書はWaterDropの基本設計書である。WaterDropとはWEBベースの出退勤管理サービスである。WaterDropを使用することで、今までタイムカードなどの紙ベースで社員の出退勤を管理していたものを電子化し、勤怠管理や給与計算など、今まで人が行っていた作業を自動化する。

# 目的

RPAで業務を自動化する以前にIT化されていない企業がまだまだ日本には残っている。IT化されていない企業が最初に取り組むべきことは、紙ベースで管理している出退勤システムを導入することである。本システムを無料で導入してもらうことでIT化のメリットを感じてもらうことが目的。

# 用語の定義

本章で使用する用語の定義について記す。

表 . 用語の定義

|  |  |  |
| --- | --- | --- |
| # | 用語 | 説明 |
|  | システム管理者 | WaterDropのインストールや設定を行う人。 |
|  | 承認者 | 社員の出退勤や交通費を承認する人。 |
|  | 社員 | 出退勤情報や交通費の申請を行う人。 |
|  | 承認依頼 | データ |
|  | 承認 | 承認はシステムに対してコミットすることを指す。  コミット後は、社員はデータを変更できない。 |
|  | 却下 | 承認依頼を拒否する場合は却下。 |
|  | プロジェクト番号 | 作業 |
|  | 社員番号 | 社員に割り振られた、会社内でユニークとなるIDのこと。WaterDropでは英数字を20文字まで指定できる。 |
|  | 基本出勤日 | 祝日，夏季休暇，年末年始，設立記念日などを除く，会社が定める勤務日のこと。平日のこと。通常は月から金が相当する。 |
|  | 基本休日 | 祝日，夏季休暇，年末年始，設立記念日などを除く，会社が定める休日のこと。通常は土日が相当する。 |
|  |  |  |
|  |  |  |

# システム構成

WaterDropはLAMP構成である。DBにmySQL，WEBフレームワークにdjango，WEBサーバにnginxを採用する。サーバはubuntu16.04以降，またはAmazonLinux2に対応する。

django

nginx

mySQL

mySQL5.7

django2.1.3

python3.6以降

nginx

ubuntu16.04

or

AmazonLinux2

図 4.1 WaterDropのシステム構成

Docker対応などは別途検討する。

ここはAWSの設定や会社ごとにEC2インスタンスを作成したり，テナント型にしたりする話を記載する。

# 運用

この章はチュートリアルのイメージ。

WaterDropの運用イメージを図 5.1に記す。まず，WaterDropはLampサーバを構築し，WaterDropを配置する。配置完了後，WaterDropの設定を行う。WaterDropの設定では社員情報の登録や，勤務カレンダー，部課情報などを登録する。WaterDropの設定が完了したあと，運用フェーズに入る。運用フェーズでは全社員が日々勤務情報を登録していく。しばしば社員が増えたり，退職したりすると，その都度WaterDropの設定で社員情報を変更する。そして，会社の締め日に給与計算を行う。

LAMPサーバの構築

WaterDropの設定

勤務情報の入力

給与計算

毎月

社員の追加，削除など

図 5.1 WaterDropの運用フロー

## LAMPサーバの構築

ここにインストール手順などを書く。できれば自動化。

**[ubuntu16.04]の場合**

**[AmazonLinux2]の場合**

## WaterDropの設定

**[社員情報の登録]**

社員番号などを追加する。

**[CSVでインポートする]**

ここは結構長くなりそう

## 勤務情報の入力

タイムカードの画面の設定が決まったら書く。チュートリアルな。

## 給与計算

ここには給与計算のチュートリアルを書く。

# 機能

WaterDropの機能一覧を表 6.1に記す。

表 . 機能一覧

|  |  |  |
| --- | --- | --- |
| # | 機能 | 概要 |
|  | ユーザ認証機能 | 社員の情報を管理する。 |
|  | タイムカード管理機能 | 出退勤のデータを入力したり，作業内容を入力したりする。 |
|  | 交通費管理機能 | 交通費申請を行う。 |
|  | ワークフロー管理 | 入力者とは異なる社員が，社員の入力したデータに誤りがないか確認する機能。 |
|  | 作業内容管理 | 1日の作業内容を管理する。 |
|  | 勤務区分管理 | 本社や支社ごとに異なる勤務時間などを管理する。 |

＃機能は機能だが，どちらかというとオブジェクト指向的に考えたほうが書きやすいかもしれない。

## ユーザ認証機能

WaterDropにアクセスするにはユーザ認証を行う。これは入力しているユーザごとにアクセスできるデータを管理するためである。ユーザアカウントは表 6.2に記す情報を持つ。

表 . ユーザ情報

|  |  |  |
| --- | --- | --- |
| # | 項目 | 説明 |
|  | ユーザ名 | ログインする際のユーザ名。基本的には社員番号を入力する。ユーザ名は半角アルファベット，半角英数字，@/./+/-/\_で1文字以上150文字以内で指摘する。必須入力項目である。すでに登録されているユーザ名は指定できない。 |
|  | パスワード | ８文字以上の150文字以内の半角文字(特殊文字も可)を指定する。ログインする際のパスワードとなる。必須入力項目である。 |
|  | 名前 | WaterDrop上での表示名である。基本的に社員の名前を150文字以内で入力する。省略できる。同性同名などで重複してもよいが，運用時に管理者が混乱する可能性があるため，なるべく重複しないようにしたほうがよい。 |
|  | メールアドレス | 社員に連絡するためのメールアドレス。150文字以内の半角英数字，@,-,.を指定できる。他の社員と重複してもよいが，なるべく異なるメールアドレスを使用したほうがよい。 |

ユーザ情報はシステム管理者が一度作成する。

ユーザを作成する際，パスワードを設定するが，システム管理者が作成するパスワードは仮パスワードとなる。仮パスワードは表 6.2#4に記すメールアドレスにメールで通知する。メールには，仮パスワードを変更するためにWaterDropが一時的に生成したログインURLとWaterDropにログインするためのユーザ名と仮パスワードが記述されている。

メールを受け取った社員はメール中のURLに，ユーザ名と仮パスワードでログインし，仮パスワードを変更する。仮パスワードの変更が完了したら，WaterDropの正規ユーザとなり，出退勤などが操作できるようになる。フローを図 6.2に記す。

システム管理者がユーザを作成する

WaterDropが社員に仮パスワードを

メールで通知する

社員が仮パスワードでWaterDropにログインし，パスワードを変更する

社員がデータを登録する

図 6.2 社員情報登録時のワークフロー

**[ユーザの状態]**

**■仮パスワードユーザ**

システム管理者がユーザを作成した状態を**仮パスワード状態**と呼ぶ。仮パスワードで操作できるのは，該当ユーザのパスワード変更だけである。

仮パスワードのメールをWaterDropが通知して，以下の条件のうちどれか一つに該当した場合，**仮パスワードロック状態**となり，仮パスワードを変更できなくなる。

* 24時間以内に仮パスワードが変更されなかった場合
* 仮パスワードを10回誤って入力した場合

仮パスワードロック状態になった場合，社員はシステム管理者に連絡し，アカウントロックを解除してもらうことで仮パスワード再発行メールがユーザに再通知され，仮パスワード変更からやり直すことができる。

仮パスワード変更後，ユーザはWaterDropにログインできるようになるが，以下の条件のどれか一つに該当した場合，アカウントはロックされ，ログインできなくなる。

* 10回連続でパスワードを誤った場合(一度成功すればカウントはリセットされる)
* 180日以上ログインがなかった場合
* 90日以上同じパスワードを使用していた場合

開始

・タイムアウト

・10回誤り

ロック解除

・10回誤り

・1年ログインなし

・90日以上パスワードを使いまわす

ロック解除

図 6.3アカウントの状態遷移図

アカウントがロックされた場合，システム管理者に連絡し，ロックを解除してもらうことで再使用できる。

|  |
| --- |
| **メモ** |
| アカウントロックを行うのは，セキュリティ上の問題からである。悪意のあるユーザがシステムにログインする際，ユーザ名とパスワードを総当たりで試していけば，いずれはログインできる。アカウントロックを行うことで，総当たりに対する所要時間を非現実的な時間をかけさせることでアカウントの乗っ取りを防ぐものことが狙いである。 |

**[ユーザ認証に関するパラメータ]**

アカウントロックに至る条件の設定はシステム管理者が認証パラメータで変更できる。

|  |  |  |  |
| --- | --- | --- | --- |
| # | ユーザ | 項目 | 説明 |
|  | 仮パスワード  ユーザ | 仮パスワード有効期間 | 仮パスワードを発行してから社員が変更できる期間。デフォルトは24時間。 |
|  | 試行回数 | 許容する仮パスワード入力回数 |
|  | 正規ユーザ | 試行回数 | 許容するパスワード入力回数。一度でも成功した場合はカウントをリセットする。デフォルトは10回。 |
|  | 許容するログアウト期間 | 何日以上ログインしなかったら，アカウントロックするか。デフォルト180日。 |
|  | パスワード有効期間 | 同じパスワードを何日以上使用できるか。デフォルト90日。 |

## 出退勤管理機能

プロジェクト1

作業A

作業B

プロジェクト2

作業C

作業D

社員

時間

時間

時間

時間

合計が勤務時間

ここから後ろの記述はメモ。



## 基本機能

社員は，出勤，退勤，欠勤すると本アプリの出勤画面で以下のデータを入力する。

* **社員番号**  
  管理者に割り当てられた社員番号。社員番号は会社内でユニークとなるIDである。
* **出勤日**  
  出勤した年月日
* **出勤時刻**  
  出勤した時刻
* **勤務区分**  
  社員の所属する勤務区分。本社勤務，支社勤務，常駐先などで基本出社時刻，退社時刻，休憩時刻などが異なる。システム管理者があらかじめ作成した勤務区分から社員の該当する勤務区分を選択する。勤務区分を元に残業時間などを計算する
* **出勤ステータス**出勤，退勤，有休，代休，リモートワークなど会社が定める勤務種別を選択する。勤務種別は会社の定める勤務種別に基づいて管理者が作成する。社員は，あらかじめ作成されている勤務種別の中から選択する

## カスタマイズ

管理者は運用する前に次のデータを作成する必要がある。

**[勤務区分]**

本社勤務，支社勤務，常駐先など異なる場所で労働する場合，場所によって勤務開始時刻，終了時刻，休憩時刻，休憩回数，出勤日が異なることがある。勤務地が異なっても残業時刻や休出の管理ができるよう，勤務区分を設定する。

勤務区分では次の内容を設定する。

・勤務区分名（本社勤務，札幌支社勤務など）

・基本出社時刻

・基本退社時刻

・勤務カレンダー

・月曜日から日曜日までの基本出勤日，基本休日

**[カレンダー]**

カレンダーは，基本休日を除く，会社が定める休日(年末年始，夏季休暇，設立記念日など)を設定する。本社，支社で異なることもあり得るため，カレンダーは複数作成できる。カレンダーは勤務区分の設定で一つだけ選択する。

カレンダーでは次の内容を設定する。

・日付

・休暇名

**[会社休出勤種別]**

会社休出勤種別はカレンダーに設定する会社全体の休出勤種別である。会社休出勤種別に設定する例としては，夏季休暇，年末年始，会社の設立記念日，震災などがあげられる。

本テーブルを作成する目的は，休暇の種別によって給料が支払われる，支払われないなどを決定するための情報となる。

**[個人休出勤ステータス]**

個人出勤種別は，タイムカードに設定する個人的な休出勤種別である。個人休出勤種別に設定する例としては，出勤，有休，慶弔休暇など個人に紐づく休暇があげられる。

本テーブルを作成する目的は，休暇の種別によって給料が支払われる，支払われないなどを決定するための情報となる。

# データ設計



## タイムカードテーブル

社員が出勤，退勤，または有休などで休んだ場合など，一日ごとの出勤状況をタイムカードテーブルで管理する。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性 | 意味 | 制約 |
|  | id | タイムカードのID。 | 主キー |
|  | employee\_id | 社員番号 | 社員テーブル.社員番号の外部キー。社員番号と年月日でユニークとなる。 |
|  | date | 年月日 | 社員番号と年月日でユニークとなる。 |
|  | inTime | 出勤時刻。休みの場合はNULL。 |  |
|  | offTime | 退勤時刻。休みの場合はNULL。 |  |
|  | remark | 備考。 |  |
|  | approval | 承認ステータス。承認は一日単位で管理者が行う。 |  |
|  | status | 出勤，有休，慶弔など日単位で勤務内容を選択する。 | 個人出勤ステータステーブルの外部キーとなる。 |
|  | workClass | 一日の入力がどの勤務区分（本社勤務，派遣先など）を選択する | 勤務区分テーブルの外部キー。 |

# 機能詳細

本章ではWaterDropの機能について記す。機能一覧を表 9.1に記す。

表 . 機能一覧

|  |  |  |
| --- | --- | --- |
| # | 機能名 | 参照先 |
|  | 認証機能 |  |
|  | 権限管理機能 |  |
|  | 出退勤データ管理機能 |  |
|  | 交通費管理機能 |  |
|  | システム管理機能 |  |



## 認証機能

ユーザがWaterDropにアクセスするには必ずユーザ名とパスワードで認証を行い，WaterDropに登録されているユーザだけが本システムを使用できる。ユーザ情報は，システム管理者が登録する。登録するユーザ情報を表 9.2に記す。

表 . ユーザ情報

|  |  |  |  |
| --- | --- | --- | --- |
| # | データ | 説明 | 備考 |
|  | 社員番号 | 会社内でユニークとなるID。英数字２０文字以内で指定する。 | 省略不可。 |
|  | パスワード | 英数字，特殊記号で8文字以上，64文字以内で指定する。 | 省略不可。 |
|  | 社員名 | 全角文字，または英字で64文字以内で指定する。 | 省略不可。 |
|  | 権限 | 社員、承認者、システム管理者のいずれかを設定する。 | 省略した場合，デフォルトで社員の権限を設定する。 |

社員は，WaterDropに社員番号とパスワードでログインする。

誤ったパスワードを指定し，一定回数ログインに失敗した場合は一定時間アカウントロックを行う。また，ユーザが一定期間パスワードを更新していない場合もアカウントロックを行う。失敗回数，アカウントロックの期間，パスワードの更新頻度はシステム管理機能で変更できる。

## 権限管理機能

WaterDropでは，３種類のユーザを前提とする。WaterDropで扱うユーザの種類と権限について表 9.3に記す。

表 . ユーザの種類と権限

|  |  |  |
| --- | --- | --- |
| # | ユーザの種類 | 権限 |
|  | 社員 | 出退勤や交通費申請を行える権限。 |
|  | 承認者 | 社員の出退勤や交通費を確定させる権限を持つ。 |
|  | システム管理者 | WaterDropの設定などすべてのデータにアクセスする権限を持つ。承認者，社員の権限はシステム管理者が割り当てる。 |

権限の包含関係を図 9.1記す。システム管理者は，承認者の上位の権限にあたり，承認者，作業者と同じことができる権限を持つ。承認者は作業者の上位の権限にあたり，作業者ができることは承認者もできる，というように包含関係で構成される。

システム管理者

承認者

社員

図 8.1　権限の包含関係

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** |  | **システム管理者** | **承認者** | **社員** |
|  | アカウントロックの設定 | 〇 | × | × |
|  | ユーザの登録・追加・削除 | 〇 | × | × |
|  |  |  |  |  |
|  |  |  |  |  |
|  | 交通費の参照 |  | △ |  |
|  | 交通費の承認・却下 | 〇 | △ | × |
|  | 出退勤の参照 | 〇 | △ |  |
|  | 出退勤の承認・却下 | 〇 | 〇 | × |
|  | 交通費の申請 | 〇 | 〇 | 〇 |
|  | 出退勤の入力 | 〇 | 〇 | 〇 |

【凡例】

〇：操作できる

△：参照権限の与えられた社員にだけアクセスできる

×：操作できない

## ログイン機能

システム管理者はWaterDropインストール時にシステム管理者のユーザ名とパスワードを設定する。WaterDropにログインする際は，設定したユーザ名とパスワードでログインする。

承認者，社員はシステム管理者がアカウントを発行する。アカウントはシステム管理者がアカウント設定を行う際，仮パスワードを発行する。仮パスワードはメールで社員に通知する。

社員はメールの支持に従い，パスワード変更完了後，WaterDropにログインできるようになる。＃システム管理者も見れないデータがない限り，仮パスワード発行に意味がないのだが・・・。

|  |
| --- |
| アカウントロックについて |
| システム管理者のアカウントがロックされた場合，WaterDropを修正できなくなってしまう。その場合は，AMIにログインし，WaterDropが使用しているDBを変更することで回復する運用とする。 |

仮登録と本登録メール切り替えは，DBのis\_active属性で管理する。

認証の手順は以下とする。

1. システム管理者が社員のアカウントを仮パスワードで作成する。WaterDrop内のユーザステータスは仮パスワードユーザとする
2. WaterDropが社員に対して仮パスワードのメールを送信する
3. 社員が社員番号と仮パスワードでログインする
4. 社員がパスワードを変更する。WaterDrop内のユーザステータスを正規ユーザに変更する

## 出退勤データ管理機能

出退勤データ管理機能では，従来タイムカードで管理していた出退勤データを電子化し，管理する。出退勤を管理することで，社員の給与計算を行うことができる。また，社員の行った作業がどのプロジェクトに関する作業で，かつ，どのような作業を行ったかも管理できる。

出退勤データ管理機能は以下の機能で構成される。

|  |  |
| --- | --- |
| # | 機能 |
|  | 社員データ管理機能 |
|  | タイムカード機能 |
|  | 作業内容管理機能 |
|  | 有休管理機能 |
|  | カレンダー機能 |
|  | 勤務区分管理機能 |
|  |  |



### 社員データ管理機能

WaterDropでは，に記す内容を社員データとして管理する。社員データはシステム管理者が設定する。社員データの内容を表 9.2に記す。

|  |
| --- |
| 設計メモ |
| クラウドで管理するなら２段認証があるほうがよいが，とりあえず今はやらない、やれない。SSLでユーザ名とパスワードをガードするので盗まれない。でもアタックは防げないからアカウントロックで耐える。 |

### タイムカード機能

タイムカード機能では社員の出退勤の情報を管理する。出退勤には日付，社員番号，作業開始日時，作業終了日時を管理する。作業開始時間，作業終了時間は分単位で指定する。休んだ場合は，社員が後日出社したときに有休，または欠勤などを指定する。ただし，会社が定める休暇の場合はシステムが休暇として扱う。会社が定める休日に関してはカレンダー機能を参照。

秒は管理しない（切り捨て）。本機能と，作業内容管理機能を連携させることで，より詳細に作業内容を管理することができる。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 操作 | 社員 | 承認者 | システム管理者 |
|  | 出退勤の登録 | 〇 | × | 〇 |
|  | 出退勤の編集 | 〇 | × | 〇 |
|  | 承認依頼 | 〇 | × | 〇 |
|  | 承認 | × | 〇 | 〇 |
|  | 却下 | × | 〇 | 〇 |

作業者が入院などで有休を設定できない場合はどうするか？

勤務時間には同じ日に作業が終わる日勤と，日をまたいで勤務する夜勤がある。

また日勤でも緊急時で２４時間以上労働する場合もある。この場合も日またがりは発生する。

### 有休管理機能

各社員が取得できる有休の回数を管理する。有休には午前半休，午後半休，年休の３種類が存在する。各有休の使用回数の上限は会社のルールに応じて上限を更新することができる。ルールはシステム管理者が設定する。

### 作業内容管理機能

WaterDropでの作業についての考え方を説明する。社員は一つ以上のプロジェクトに所属する。社員は所属するプロジェクトに応じた作業を行う。作業には複数の種類があり，社員は一日のうちに一つ以上の作業を行う。時間は一つ一つの作業に割り振られ，合計時間が１日の勤務時間となる(図 9.2)。

図 8.2 WaterDropの作業の考え方

### 給与明細作成機能

## 交通費管理機能

交通費関連の機能について説明する。

## システム管理機能（ここのボリュームがやばいことになりそう）

システム管理機能は，勤怠や交通費を管理する上で必要な設定や，トラブル発生時にデータを修復したり，DBのバックアップを行ったりする機能である。システム管理機能に属する機能はすべてシステム管理者だけが設定できる。

ユーザ管理機能とかログ機能とかカレンダー設定機能とか権限設定機能とか。



### カレンダー作成機能

複数の事務所で会社を運営している場合，事務所によって休日の区分が異なる場合がある。事務所単位にカレンダーを作成できる機能である。システム管理者が作成する。

カレンダー設定テーブルにカレンダー名と基本休日（大抵は土日）を選択する。

### インポート機能

DBの設定をCSVなどから一括でインポートする機能。インポートできるテーブル、できないテーブルを明記すること。

### エクスポート機能

DBのレコードをエクスポートする機能。エクスポートできるテーブル、できないテーブルを明記すること。

### バックアップ機能

データ破壊に対してDBをバックアップする機能。

## RPA抑止機能

ベイシスのRPAしかアクセスできないようにしておく機能。

# 付録



## djangoでメールを送信する

settings.pyを変更する。

|  |
| --- |
| EMAIL\_HOST = 'smtp.gmail.com'  EMAIL\_PORT = 587  EMAIL\_HOST\_USER = 'gmailアカウント名'  EMAIL\_HOST\_PASSWORD = 'gmailパスワード'  EMAIL\_USE\_TLS = True |

メール関数を作成する。赤字でwaterDropAppのsettings.pyが反映される謎仕様

|  |
| --- |
| from django.conf import settings  def mail3(request):  subject = "題名"  message = "本文\\nです"  from\_email = settings.EMAIL\_HOST\_USER  recipient\_list = [  "narito@mail.torina.top"  ]  send\_mail(subject, message, from\_email, recipient\_list)  return render(request, 'testapp/index.html') |

本来ならば、この設定だけでgmailは送信できるがgoolgleのセキュリティで信頼性の低いアプリケーションからのログインを許可する、の設定がないとNG。

社員の出退勤を管理できる

休日出勤の計算

基本労働時間の設定

残業時間の計算

カレンダーによる休日の設定

有休の計算

承認ルートの作成

給料計算？

作業コードの作成

詳細コードの設定とか

報告書の作成

出退勤管理機能

交通費管理機能

休日設定機能

ユーザ管理機能

# メモ

androidに対応するのは大有り。10万ダウンロード超えている模様。

あとは完璧なものを作るというより、育てられるシステムを視野に入れておくこと。

# デプロイ

WaterDropは，提供するサーバ上からクリックするだけで使用できる。

# URL設計

以下は、こんな設計が必要だよという例。

