

配達支援システム 内部設計書

第0版
ONO-Systems

平成30年12月4日

目 次

1	開発対象のシステム概要	2
2	開発環境	2
3	動作環境	2
4	コーディング規約	2
4.1	コンポーネントやクラスについて	2
4.2	変数の命名について	3
4.3	関数の命名について	3
4.4	コーディングについて	3
4.5	コメントについて	4
5	ネットワーク設計	4
6	Android モジュール設計	5
6.1	モジュール構成	5
6.2	モジュール仕様	5
7	Server モジュール設計	5
7.1	モジュール構成	5
7.2	モジュール仕様	5
8	データベース設計	5
8.1	各テーブルの詳細	5
8.1.1	消費者テーブル	5
8.1.2	配達者テーブル	5
8.1.3	商品テーブル	6
8.1.4	管理者テーブル	6
8.1.5	地図テーブル	6
9	バージョン管理規約	7

1 開発対象のシステム概要

本システムは、配達物の配達支援を行うシステムです。主な機能を以下に示します。

- 利用者への通知機能
- 配達員の位置情報の表示機能
- 利用者の選択結果のリアルタイム表示機能
- 音声読み上げ機能

2 開発環境

本システムの開発環境を以下に示します。

- Android アプリケーション
Android Studio ver4.2 とか
- サーバ
AWS(AmazonWebService)
- 開発言語
Java, MySQL
- 文書・コード管理
GitHub

3 動作環境

- Android アプリケーション
Android 4.2 とか
- サーバ
AWS(AmazonWebService)

4 コーディング規約

本プロジェクトのプログラムは、以下の規則を遵守します。

4.1 コンポーネントやクラスについて

- UpperCamelCase を利用する
- Component 名は最後につける (Activity, Fragment, TextArea など)
- 本プロジェクトで頻繁に利用する名詞は以下の表を用いて命名する (表 1 参照)

表 1: 本プロジェクトで利用する名詞の英単語表

外部設計書での名前	英単語
アカウント保持者	User
利用者	Customer
配達員	Courier
配達物	Delivery

4.2 変数の命名について

- 命名には英語を用いる
- 名前から役割が読み取れるように命名する
- グローバル変数に用いる英単語は省略しない (String ⇒ Str など)
- LowerCamelCase を利用する
- 定数は全て大文字で表し、スネークケースで連結する
- for や while のカウンタ変数には、i,j,k を用いる

4.3 関数の命名について

- 意味と英単語の対応付けを統一する (表 2)
- 名前は動詞で始める
- LowerCamelCase を利用する

表 2: 命名に使う英単語表

意味	英単語	例
真偽値を取得	is	#isCreated
値を代入	set	#setDelivery
引数を含むか判定	has	#hasFragment
特定の動作をしたときに動作	on	#onClick
新しく作る	create	#createSubActivity
新しく作る	new	#newDelivery
更新	update	#updateDeliveryDate
サーバから情報を取得	fetch	#fetchDeliveryList

4.4 コーディングについて

- 字下げは半角スペース 2 つを用いる
- マジックナンバーは使用しない

- 安易にネット上のソースコードを利用しない
- class や if、while などのブロック始点のブラケット ('{') は改行せずに、半角スペースを空けて記述する
- 機能が完成したとき (Pull Requests を出すとき) にデバッグ用のコードを残さない

4.5 コメントについて

- 複雑な処理はコメントをつける
- メソッドにはコメントをつける
- TODO:など、コードの説明でないコメントは PullRequests を出すときには消す

5 ネットワーク設計

本システムのネットワークは図1のように構成されます。

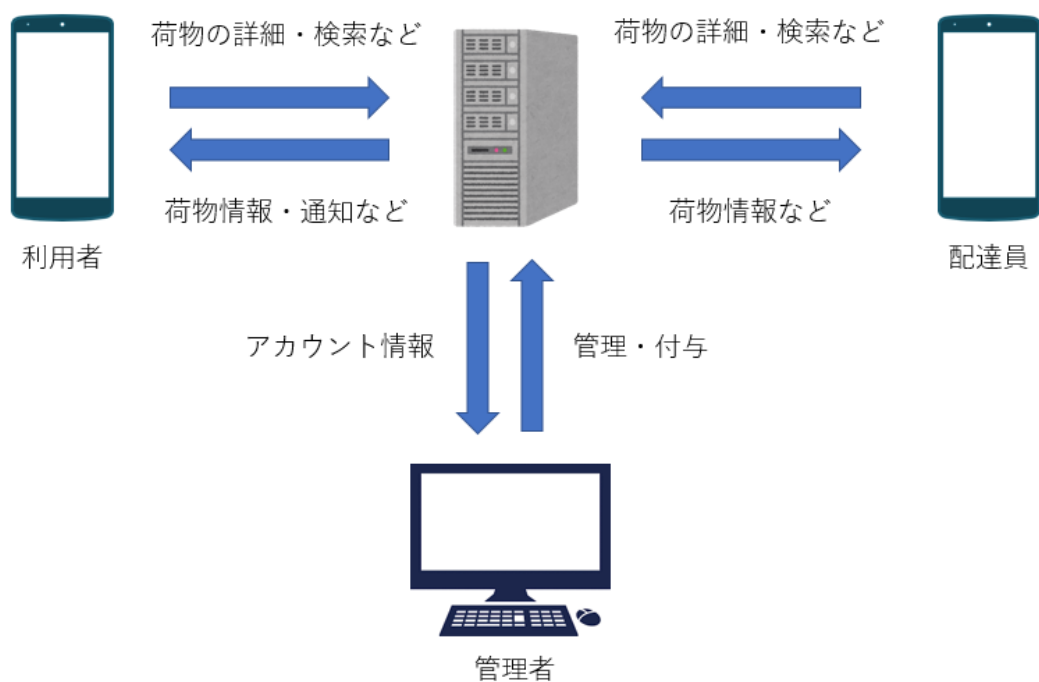


図 1: ネットワーク構成図

6 Android モジュール設計

6.1 モジュール構成

6.2 モジュール仕様

7 Server モジュール設計

7.1 モジュール構成

7.2 モジュール仕様

8 データベース設計

本システムではデータベースに AWS(AmazonWebService) を使用します。
ER 図など

8.1 各テーブルの詳細

本システムのデータベースには、5 個のデータテーブルを用います。各データテーブルの役割と属性を以下に示します。

8.1.1 消費者テーブル

消費者テーブルでは、消費者に関する情報を管理します。このテーブルのデータテーブルを表 3 に示します。

表 3: 消費者テーブル

属性	データ型/長	NULL	Key	初期値	その他
customer_id	int(9) unsigned	NO	PRIMARY	NULL	auto_increment
name	varchar(64)	NO		NULL	
mail	varchar(64)	NO		NULL	
tel	varchar(11)	YES		NULL	
address	varchar(128)	YES		NULL	
hash	varchar(255)	NO		NULL	
salt	varchar(255)	NO		NULL	

8.1.2 配達者テーブル

配達者テーブルでは、配達者に関する情報を管理します。このテーブルのデータテーブルを表 4 に示します。

表 4: 配達者テーブル

属性	データ型/長	NULL	Key	初期値	その他
driver_id	int(9) unsigned	NO	PRIMARY	NULL	auto_increment
name	varchar(64)	NO		NULL	
mail	varchar(64)	YES		NULL	
tel	varchar(11)	YES		NULL	
store_code	varchar(64)	YES		NULL	
account_type	int(1) unsigned	YES		NULL	
hash	varchar(255)	NO		NULL	
salt	varchar(255)	NO		NULL	

8.1.3 商品テーブル

商品テーブルでは、商品に関する情報を管理します。このテーブルのデータテーブルを表 5 に示します。

表 5: 商品テーブル

属性	データ型/長	NULL	Key	初期値	その他
slip_number	varchar(12)	NO	PRIMARY	NULL	
name	varchar(64)	NO		NULL	
address	varchar(128)	NO		NULL	
time	varchar(11)	YES		NULL	
delivery_status	int(1) unsigned	YES		NULL	
receive_status	int(1) unsigned	YES		NULL	
customer_id	int(9) unsigned	NO	MULTIPLE	NULL	
driver_id	int(9) unsigned	NO	MULTIPLE	NULL	

8.1.4 管理者テーブル

管理者テーブルでは、管理者に関する情報を管理します。このテーブルのデータテーブルを表 6 に示します。

8.1.5 地図テーブル

地図テーブルでは、地図に関する情報を管理します。このテーブルのデータテーブルを表 7 に示します。

表 6: 管理者テーブル

属性	データ型/長	NULL	Key	初期値	その他
manager_id	int(1) unsigned	NO	PRIMARY	NULL	auto_increment
name	varchar(64)	NO		NULL	
mail	varchar(64)	YES		NULL	
store_code	varchar(64)	YES		NULL	
account_type	int(1) unsigned	YES		NULL	
hash	varchar(255)	NO		NULL	
salt	varchar(255)	NO		NULL	

表 7: 地図テーブル

属性	データ型/長	NULL	Key	初期値	その他
map_id	varchar(12)	NO	MULTIPLE	NULL	
lat	double(8,6)	YES		NULL	
lng	double(9,6)	YES		NULL	

9 バージョン管理規約

本システムの開発では、Github を用いてファイルの管理を行います。Github を使用する際には、以下の規則を遵守します。

- ドキュメント関連の資料は、onosystem-doc で管理する
- Android のソースコードは、onosystem-android で管理する
- Server のソースコードは、onosystem-server で管理する
- 編集作業を行う際には、ブランチを切ってコミットする
- 開発用ブランチの名前は、「dev_〇〇（開発している機能名）」にする
- 細かい頻度でコミットする（1日の作業ごとに纏めてコミットしない）
- コミットのコメントはわかりやすい内容にする
- Pull Requests されたものを確認し、評価をリアクションのアイコンを追加することで示す
- Pull Requests に対して高評価が3つ以上ある場合には master に Merge する