

TERASOLUNA Server Framework for Java 5.x

外部設計作業要領・規約（雛型）

第1.1.1版

2015年09月04日

**変更履歴**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版数** | **制改訂日** | **変更箇所** | **変更内容** | **変更者** |
| 1.1.1 | 2015年09月04日 |  | 初版制定 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

##### カスタマイズポイント一覧

プロジェクトごとに修正を検討する必要のあるカスタマイズポイントについて、以下の表に記述する。

|  |  |
| --- | --- |
| **カスタマイズポイント** | **説明** |
| 2. ソフトウェア構成 | システムの構成に合わせて変更する。 |
| 3. 開発環境 | プロジェクトで使用するものに合わせて変更する。プロジェクト内でツールの種類やバージョンが統一されていないと、ファイル破壊やデータ損失の恐れがあるため、必ず規定し統一をはかること。 |
| 4.1. 外部設計の作業の流れ | "ユースケース"や"イベント"などの用語はプロジェクトの標準に合わせて変更して構わない。例えば、ユースケースは"機能"、イベントは"処理"と読み替えるパターンが考えられる。用語を変更した場合は成果物名や設計書中の項目名も併せて変更すること。 |
| 4.3. 作業概要と成果物 | プロジェクトの特性に合わせて成果物の種類を変更する。例えば、本書では帳票機能を対象外として関連する成果物を省略しているため、必要に応じて追加すること。 |
| 5. 命名付与基準 | TERASOLUNA Server Framework for Java 5.x の制約として規定している命名付与基準はないため、各プロジェクトの標準に合わせて変更して構わない。 |
| 6.2. ログメッセージ | プロジェクトの特性に合わせてレベルやカテゴリを変更する。本書に記載した以外の代表的なログとしては、外部システムとの通信におけるエラー解析を目的とした「外部通信ログ」や、業務プロセスの実行の記録を目的とした「監査ログ」などがある。 |
| 6.3.2 チェック実施場所 | 入力チェックをクライアント側でも行う場合、対象とするチェックの範囲（単項目チェックのみ行うのか、相関項目チェックまで行うのか）を検討すること。相関項目チェックまで行うと、JavaScriptの個別実装が必要となるため、開発コストを加味して方針を決めること。 |
| 6.3.5 単項目チェックのルール | 一覧に存在しないチェックルールが必要な場合は追加する。併せて設計書（画面仕様の画面項目シート）のフォーマットも変更すること。 |

|  |  |
| --- | --- |
| **カスタマイズポイント** | **説明** |
| 6.5. 排他制御 | プロジェクトの特性に合わせて、短時間トランザクション、長時間トランザクションについてそれぞれ、悲観的ロックや楽観的ロックといった排他制御の方式について記述すること。長時間トランザクションについては、そもそも排他制御を行わない選択も考えうる。 |
| 6.9. 国際化 | 国際化対応を行う際は、対応する言語の種類を明記し、画面仕様や画面メッセージを設計する際の方針を追加すること。 |
| 6.11. ファイルアップロード、ダウンロード | システムの対象機能にファイルアップロード、ダウンロードが含まれる場合は、記述を追加すること。ファイルアップロードでは、複数ファイルのアップロードを許容するのか、アップロード時に実施する検証の方針などを記述する。ファイルダウンロードでは、ダウンロード後の動作の方針（ファイルを保存させるのか、画面に表示させるのか）、デフォルトのファイル名の方針などを記述する。 |
| 6.12. 帳票 | システムの対象機能に帳票が含まれる場合は記述を追加すること。設計方針と併せて成果物の追加や設計書の記述要領も追記すること。 |

[1. はじめに 7](#_Toc415045769)

[1.1. 目的 7](#_Toc415045770)

[1.2. 留意事項 7](#_Toc415045771)

[1.3. 章構成 7](#_Toc415045772)

[2. ソフトウェア概要 8](#_Toc415045774)

[2.1. ソフトウェア概要 8](#_Toc415045775)

[3. 開発環境 10](#_Toc415045776)

[4. 外部設計の流れ 11](#_Toc415045777)

[4.1. 外部設計の作業の流れ 11](#_Toc415045778)

[4.2. 外部設計の入力情報 12](#_Toc415045779)

[4.3. 作業概要と成果物 12](#_Toc415045780)

[5. 命名付与基準 13](#_Toc415045781)

[5.1. ID体系 13](#_Toc415045782)

[5.2. 成果物ファイル名 13](#_Toc415045783)

[6. 外部設計の基本方針 14](#_Toc415045784)

[6.1. 画面メッセージ 14](#_Toc415045785)

[6.2. ログメッセージ 15](#_Toc415045786)

[6.3. 入力チェック 16](#_Toc415045787)

[6.3.1 入力チェックの種類 16](#_Toc415045788)

[6.3.2 チェック実施場所 16](#_Toc415045789)

[6.3.3 チェック対象 16](#_Toc415045790)

[6.3.4 エラーメッセージの出力場所 17](#_Toc415045791)

[6.3.5 単項目チェックのルール 17](#_Toc415045792)

[6.3.6 単項目チェックのエラーメッセージ 17](#_Toc415045793)

[6.4. トランザクション範囲 18](#_Toc415045794)

[6.5. 排他制御 18](#_Toc415045795)

[6.6. ページネーション 18](#_Toc415045796)

[6.7. 二重送信防止、不正画面遷移チェック 18](#_Toc415045797)

[6.7.1 二重送信防止 18](#_Toc415045798)

[6.7.2 不正画面遷移チェック 19](#_Toc415045799)

[6.8. エラー処理 19](#_Toc415045800)

[6.9. 国際化 20](#_Toc415045801)

[6.10. 認証・認可 20](#_Toc415045802)

[6.11. ファイルアップロード、ダウンロード 20](#_Toc415045803)

[6.12. 帳票 20](#_Toc415045804)

[7. 設計書の記述要領 21](#_Toc415045805)

[7.1. メッセージ一覧 21](#_Toc415045806)

[7.1.1 記述単位 21](#_Toc415045807)

[7.1.2 記述項目 21](#_Toc415045808)

[7.2. 画面一覧 21](#_Toc415045809)

[7.2.1 記述単位 21](#_Toc415045810)

[7.2.2 記述項目 21](#_Toc415045811)

[7.3. 画面遷移図 22](#_Toc415045812)

[7.3.1 記述単位 22](#_Toc415045813)

[7.3.2 記述項目 22](#_Toc415045814)

[7.3.3 記述方法詳細 22](#_Toc415045815)

[7.4. 画面仕様 24](#_Toc415045816)

[7.4.1 記述単位 24](#_Toc415045817)

[7.4.2 レイアウトシート 24](#_Toc415045818)

[7.4.2.1 記述項目 24](#_Toc415045819)

[7.4.2.2 記述方法詳細 24](#_Toc415045820)

[7.4.3 画面項目シート 24](#_Toc415045821)

[7.4.3.1 記述項目 24](#_Toc415045822)

[7.4.3.2 記述方法詳細 24](#_Toc415045823)

[7.4.4 画面状態シート 24](#_Toc415045824)

[7.4.4.1 記述項目 24](#_Toc415045825)

[7.4.4.2 記述方法詳細 24](#_Toc415045826)

[7.5. イベント一覧 24](#_Toc415045827)

[7.5.1 記述単位 24](#_Toc415045828)

[7.5.2 記述項目 24](#_Toc415045829)

[7.5.3 記述方法詳細 24](#_Toc415045830)

[7.6. イベント仕様 24](#_Toc415045831)

[7.6.1 記述単位 24](#_Toc415045832)

[7.6.2 記述項目 24](#_Toc415045833)

[7.6.3 記述方法詳細 24](#_Toc415045834)

# はじめに

## 目的

外部設計作業要領・規約は、システム開発に携わる業務開発者に対し、外部設計の一連の作業手順、設計の観点、遵守すべき規約を示すことで、作業の効率化を図るとともに、多人数での開発における作業者間の品質のばらつきを防止することを目的としている。

なお、本書ではサンプルの前提として架空の「航空チケット予約システム」を題材にしている。「航空チケット予約システム」のシステム概要については、TERASOLUNA Server Framework for Java 5.x 開発手順で別途提供している「サンプルアプリケーションドキュメント」を参照されたい。本書の本文中に登場する「本システム」とは、特に断りのない限りこの「航空チケット予約システム」を指す。

## 留意事項

本書は外部設計作業要領・規約のサンプルであり、プロジェクト特性に合わせてテーラリングされることを前提としている。本書を利用する際は、「カスタマイズポイント一覧」で提示している情報を参考にして、記述内容の追加、変更、削除を行うこと。

## 章構成

本書の章構成をに示す。

表 1外部設計 作業要領・規約の章構成

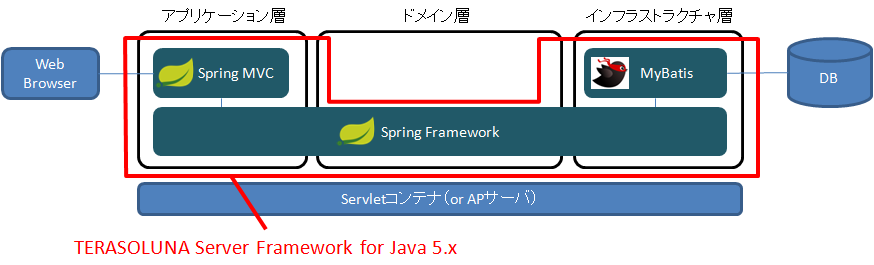
|  |  |  |
| --- | --- | --- |
| 章番 | 章名 | 概要 |
| 1 | はじめに | 本章  外部設計 作業要領・規約の目的、留意事項、および章構成について記載 |
| 2 | ソフトウェア概要 | 使用するソフトウェアの概要について記載 |
| 3 | 開発環境 | 外部設計工程の成果物作成に使用するツール等の環境について記載 |
| 4 | 外部設計の流れ | 外部設計工程の作業の流れ、前提となる入力情報、各作業で作成する成果物とその作成単位について記載 |
| 5 | 命名付与基準 | 「」で記載した成果物を作成する上で遵守する必要のある、ID体系や成果物ドキュメントの命名付与基準について記載 |
| 6 | 外部設計の基本方針 | 「」で記載した作業の流れに従い、各作業で考慮する必要があるポイントを記載 |
| 7 | 設計書の記述要領 | 「」で記載した成果物である設計書の記述単位および記述項目について記載 |

# ソフトウェア概要

本章では、本システムのソフトウェア概要について説明する。

## ソフトウェア概要

本システムで使用するソフトウェアの概要をに示す。



1. 図 1　ソフトウェア概要図

本システムで採用するTERASOLUNA Server Framework for Java 5.x で使用するフレームワークの一覧[[1]](#footnote-1)をに示す。

1. 表 2 使用フレームワーク一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **Ver** | **説明** | **ライセンス** | **URL** |
| Spring Framework | 4.1.4 | DIコンテナ、AOPなどの機能を持つオープンソースフレームワーク。 | Apache License 2.0. | http://spring.io/ |
| Spring MVC | 4.1.4 | 画面遷移、入力チェックなどのプレゼンテーション層の処理を行なうオープンソースフレームワーク。 | Apache License 2.0. | http://spring.io/ |
| MyBatis | 3.2.8 | SQLをベースとした、O/Rマッピングフレームワーク。 | Apache License 2.0. | http://mybatis.github.io/mybatis-3/ |

# 開発環境

本章では、外部設計の成果物作成に使用するツール等の開発環境をに示す。

1. 表 3 開発環境一覧

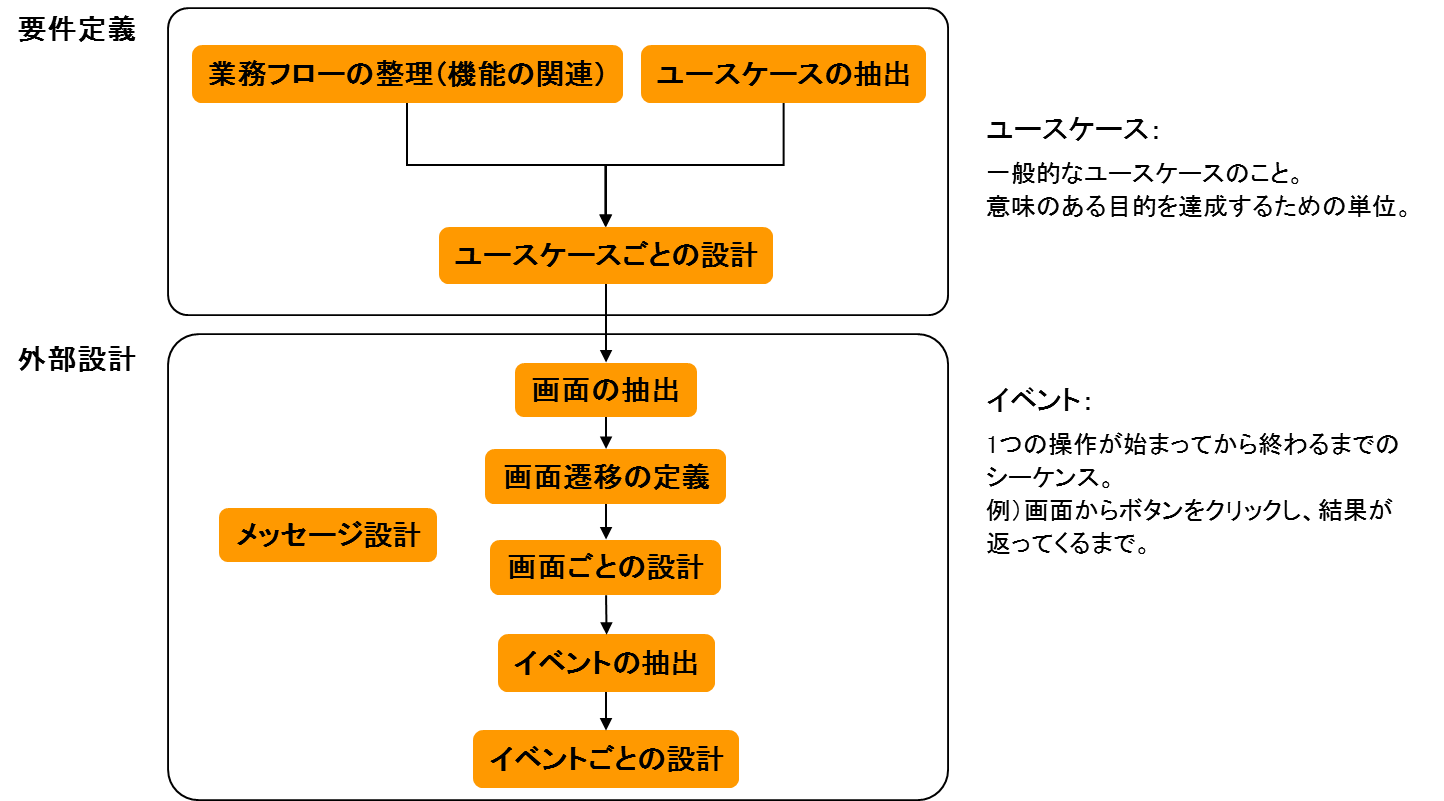
|  |  |
| --- | --- |
| **名称** | **説明** |
| Microsoft Office Excel 2010 | 表計算・文章作成ツール。外部設計の成果物は基本的にExcel形式で作成する。 |
| Astah\* community 6.7 | UMLモデリングツール。画面遷移図を作成する際はUMLのアクティビティ図で作成し、Excel形式の設計書に貼付する。 |

# 外部設計の流れ

本章では、外部設計の作業の流れ、前提となる入力情報、各作業で作成する成果物とその作成単位について示す。

## 外部設計の作業の流れ

外部設計で業務開発者が実施する作業の流れをに示す。前工程からの流れが分かるよう、要件定義工程の作業も併せて記載している。



1. 図 2　外部設計の作業の流れ

## 外部設計の入力情報

外部設計で業務開発者が実施する作業の入力情報の一覧をに示す。

表 4　外部設計の入力情報一覧

|  |  |  |  |
| --- | --- | --- | --- |
| **入力情報** | **概要** | **作成工程** | **作成主幹** |
| 要件定義書 | 要件定義工程で作成した各種定義書。 | 要件定義 | 全チーム |
| UI規約[[2]](#footnote-2) | 画面を作成するうえで基準となる画面遷移やレイアウトなどの規約を示すドキュメント。 | 外部設計 | AP基盤チーム[[3]](#footnote-3) |
| 外部設計作業要領・規約（本書） | 業務開発者に対し、外部設計の一連の作業手順、設計の観点、遵守すべき規約を示すドキュメント。 | 外部設計 | AP基盤チーム |

## 作業概要と成果物

外部設計で実施する作業の概要と成果物、各成果物の作成単位をに示す。各成果物の記述については7章を参照すること。

1. 表 5 作業概要と成果物一覧

|  |  |  |  |
| --- | --- | --- | --- |
| **作業** | **概要** | **成果物** | **作成単位** |
| メッセージ設計 | システムが出力するメッセージを設計する。 | メッセージ一覧 | システムで1部 |
| 画面の抽出 | 要件定義の成果物をもとにシステムで必要な画面を抽出する。 | 画面一覧 | システムで1部 |
| 画面遷移の定義 | 抽出した画面間の関係を明確にし画面遷移を定義する。 | 画面遷移図 | ユースケース単位[[4]](#footnote-4) |
| 画面ごとの設計 | 画面項目を抽出し、UI規約に従って画面のレイアウト、画面項目、画面状態を設計する。  複数のユースケースで利用される画面がある場合、画面を共通化する。 | 画面仕様 | 画面単位 |
| イベントの抽出 | 画面から発出されるイベントを抽出する。 | イベント一覧 | システムで1部 |
| イベントごとの設計 | イベントごとの処理内容、入出力項目を設計する。 | イベント仕様 | ユースケース単位 |

# 命名付与基準

本章では、外部設計を行う上で遵守すべき命名付与基準を示す。

## ID体系

ID体系の一覧をに示す。

表 6　ID体系一覧

|  |  |  |  |
| --- | --- | --- | --- |
| **項番** | **項目名** | **形式** | **例** |
| 1 | 画面ID | <ユースケースID><99> | C101 |
| 2 | イベントID | <画面ID><99> | C10101 |
| 3 | 画面メッセージID | <メッセージタイプ>.<プロジェクト区分>.<ユースケースID>.<エラーレベル><999> | e.ar.c1.2001 |
| 4 | ログメッセージID | <メッセージタイプ>.<プロジェクト区分>.<ユースケースID>.L<エラーレベル><999> | i.ar.a1.L0001 |
| 5 | 入力チェック基本メッセージID | TERASOLUNA Server Framework for Java 5.x 開発手順が定めるID体系  ※詳細は「」を参照 | - |

* + <99>：01から始まる2桁の連番
  + <999>：01から始まる3桁の連番
  + <メッセージタイプ>：”i”(info)、”w”(warn)、”e”(error)のいずれか
  + <プロジェクト区分>：”ar”固定
  + <エラーレベル>：0-1(正常)、2-4(業務エラー[準正常])、5-7(入力チェックエラー)、8（業務エラー[異常]）、9(システムエラー)のいずれか。

## 成果物ファイル名

成果物ファイル名の一覧をに示す。

表 7　成果物ファイル名一覧

|  |  |  |  |
| --- | --- | --- | --- |
| **項番** | **成果物** | **形式** | **例** |
| 1 | メッセージ一覧 | <成果物名> | メッセージ一覧 |
| 2 | 画面一覧 | 画面一覧 |
| 3 | イベント一覧 | イベント一覧 |
| 4 | 画面遷移図 | <成果物名>\_<ユースケース名> | 画面遷移図\_チケットを予約する |
| 5 | 画面仕様 | <成果物名>\_<ユースケース名>\_<画面名> | 画面仕様\_チケットを予約する\_お客様情報入力画面 |
| 6 | イベント仕様 | <成果物名>\_<ユースケース名> | イベント仕様\_チケットを予約する |

# 外部設計の基本方針

本章では、外部設計を行う上での基本方針を示す。

## 画面メッセージ

画面メッセージとは、ユーザの画面操作の結果、画面に表示する文言を指す。本システムではメッセージ内容の重要度に応じ、に示す3レベルに分けてメッセージを管理する。メッセージ定義の際は出力するメッセージが以下のどのレベルに属するか意識して定義すること。

1. 表 8 メッセージレベル一覧

|  |  |  |
| --- | --- | --- |
| メッセージレベル | カテゴリ | 概要 |
| INFO | 情報メッセージ | ユーザの操作による処理が正常に実行された後、画面に表示するメッセージ。 |
| WARN | 警告メッセージ | 何らかの問題があるが、処理を続行できる場合のメッセージ。  （例：パスワード有効期限切れが近い場合の通知メッセージ） |
| ERROR | 入力エラーメッセージ | ユーザの入力値が不正な場合に、入力画面に表示するメッセージ。 |
| 業務エラーメッセージ | 業務ロジックでのエラーと判定された場合のメッセージ。 |
| システムエラーメッセージ | システム的なエラー（データベースとの接続不可など）で、ユーザの操作により正常復帰できない場合のエラーメッセージ。 |

　　似たようなメッセージは置換文字列を使用して、１つにまとめること。

　　例）　下記の2つのメッセージを、置換文字列（{0}）を使用して、１つにまとめる。

　　　　　「目的地コードは必須入力です」

　　　　　「到着地コードは必須入力です」

　　　　　　　　　　　　　↓

　　　　　「{0}は必須入力です」

　　　　　置換文字列が２つ以上ある場合は、{0} {1} {2} ・・・とする。

## ログメッセージ

本システムではに示すカテゴリに応じ、5つのレベルに分けてメッセージを管理する。ログ出力の際はそのログが以下のどのレベルに属するか意識して定義すること。

1. 表 9　ログレベル一覧

|  |  |  |  |
| --- | --- | --- | --- |
| ログレベル | カテゴリ | 出力目的 | 運用時の出力有無 |
| TRACE | 性能ログ | リクエストの処理時間の測定 | × |
| DEBUG | デバッグログ | 開発時のデバッグ | × |
| INFO | アクセスログ | 業務量の把握 | ○ |
| 業務ログ | ログイン・ログアウトの記録 | ○ |
| WARN | 業務エラーログ | 業務エラーの記録 | ○ |
| ERROR | システムエラーログ | システム運用の継続が困難な事象の記録 | ○ |

## 入力チェック

入力チェックとは、各イベントでシステムが受け取った入力値に対して、その値がシステムで想定している形式やサイズであるかをチェックすることである。エラーがあった場合は、ユーザに対して再入力を求めるメッセージを画面に表示するなどの処理を行う。

### 入力チェックの種類

入力チェックの種類をに示す。

表 10　入力チェックの種類

|  |  |
| --- | --- |
| 種類 | 概要 |
| 単項目チェック | 項目単体で完結するチェック  ・必須チェック、桁数範囲チェック、文字種別チェック　など |
| 相関項目チェック | 複数の項目にまたがるチェック  複数の入力項目に同じ値を入力する必要のある場合  例）パスワードと確認パスワードの一致確認  複数の入力項目に別々の値を入力する必要のある場合  例）出発空港と到着空港  複数の入力項目で一つの意味のあるデータを構成する場合  例）郵便番号の上３桁と下４桁  ・複数の入力項目の間にルールが存在する場合  例）生年月日と年齢の整合性　　　　など |

### チェック実施場所

入力チェックにはサーバ側で行うチェックと、クライアント側で行うチェックがある。サーバ側でのチェックはシステムの整合性を保つ上で必須である。一方、クライアント側でのチェックはユーザビリティ向上の観点やネットワークへの不要なデータ送信を防止する等の要件があれば行う。本システムでは原則としてサーバ側でのみチェックを行なう。

### チェック対象

ユーザが画面上で文字入力する項目（テキストボックスなど）だけでなく、明示的に文字入力しない項目（リストボックス、コンボボックス、ラジオボタン、チェックボックス、隠しフィールドなど）についても、改竄などにより想定外の値が送信されることを考慮し、入力チェックの対象とすること。

### エラーメッセージの出力場所

単項目チェックのエラーメッセージは、対象の項目が分かるように項目の近く（右部や下部）に表示する。相関項目チェックのエラーメッセージは、ページ上部にまとめて表示する。単項目チェックエラーでもメッセージを項目の近くに表示させにくい場合は、ページ上部にまとめて表示してよいが、その場合はメッセージに対象の項目名を含めること。

### 単項目チェックのルール

単項目チェック時はに示すルールを適用する。これら以外の入力チェックルールが必要な場合は、AP基盤チームに申請すること。

1. 表 11　単項目チェックのルール一覧

| **ルール名** | | **概要** |
| --- | --- | --- |
| 必須 | | 入力されているか検証する。 |
| 桁数範囲 | | 文字列桁数が指定した範囲であるか検証する。 |
| 文字種別 | 半角文字 | 文字種が半角文字のみであるか検証する。 |
| 半角英数字 | 文字種が半角英数字のみであるか検証する。 |
| 半角英字 | 文字種が半角英字のみであるか検証する。 |
| 半角数字 | 文字種が半角数字のみであるか検証する。 |
| 半角カタカナ | 文字種が半角カタカナのみであるか検証する。 |
| 全角文字 | 文字種が全角文字のみであるか検証する。 |
| 全角英数字 | 文字種が全角英数字のみであるか検証する。 |
| 全角英字 | 文字種が全角英字のみであるか検証する。 |
| 全角数字 | 文字種が全角数字のみであるか検証する。 |
| 全角カタカナ | 文字種が全角カタカナのみであるか検証する。 |
| 全角＋半角文字 | 文字種が全角＋半角文字のみであるか検証する。 |
| クレジットカード番号形式 | | Luhnアルゴリズムでクレジットカード番号が妥当か検証する。  ※使用可能な番号かどうかをチェックするわけではない。 |
| Emailアドレス形式 | | RFC2822[[5]](#footnote-5)に準拠したEmailアドレス形式であるか検証する。 |
| URL形式 | | RFC2396[[6]](#footnote-6)に準拠したURL形式であるか検証する。 |
| 未来日付 | | 未来日付であるか検証する。 |
| 過去日付 | | 過去日付であるか検証する。 |
| 最小値 | | 指定した最小値以上であるか検証する。 |
| 最大値 | | 指定した最大値以下であるか検証する。 |

### 単項目チェックのエラーメッセージ

TERASOLUNA Server Framework for Java 5.x が提供する単項目チェックルールのデフォルトのエラーメッセージは英語のみであるため、日本語のメッセージを個別に定義する必要がある。メッセージを定義する際はルールごとの既定のメッセージIDを使用する。メッセージIDとデフォルトのメッセージ内容はTERASOLUNA Server Framework for Java (5.x)Development Guideline「5.5. 入力チェック」を参照のこと。

## トランザクション範囲

1イベント＝1トランザクションを原則とし、トランザクションの範囲はサーバ側の業務ロジック処理が呼び出される時点から終了するまでとする。1つのイベント仕様に記述する内容が１トランザクション内の処理となる。1トランザクション以外になる場合（エラーが発生した際にエラーログをDBに保存する場合など）は、AP基盤チームに申請すること。

## 排他制御

1リクエスト内の短時間トランザクションは、DBMSのトランザクション機能を利用した悲観的ロックを行い、複数のユーザが同時に更新することで不整合が発生しないようにする。

ユーザが要求したデータを参照し画面に表示する処理と、参照したデータをユーザが画面で修正後、DBへ更新する処理が複数リクエストにまたがるといった長時間トランザクションにおいては、想定される更新競合の発生頻度に応じ排他制御方法を使い分ける。更新競合の発生頻度が多くない業務は、楽観的ロックを利用し、DBの更新時に競合を検知し、ユーザへエラー通知する。更新競合の発生頻度が多い業務は、データ参照時に他のユーザが編集中であるかをチェックし、複数のユーザによる同時編集を防止する。

## ページネーション

ページネーションとは、DBの検索結果などの一覧表示をする際に、固定件数ごとに複数のページに分割して表示する画面デザインである。検索条件に一致するデータが大量になる場合、ページネーションを適用すること。ページネーションに対応したイベント処理では、表示するページ分のデータのみ取得するよう設計する。例えば、トータル100件で、画面に1ページ当たり10件表示する場合、ページ遷移のイベントが発生するごとに対象ページ分の10件をDBから取得する。

## 二重送信防止、不正画面遷移チェック

### 二重送信防止

Webアプリケーションでは、送信ボタンの複数回クリックや、完了画面の再読み込み等の操作を実施することにより、処理が2回実行（二重送信）されてしまうことがある。本システムでは、二重送信の防止策として、サーバサイドとクライアントサイドでそれぞれ下記の対応を行う。

* サーバサイド：PRG（Post-Redirect-Get）パターンの適用

PRGパターンとは、更新処理などの際に、POSTによるリクエストに対してリダイレクトを返し、GETのレスポンスとして遷移先の画面を表示する方法である。

PRGパターンを利用することで、更新処理の実行後、遷移先画面で更新ボタンを押下した際、POSTの更新処理ではなく、GETによる遷移先画面の再描画処理が行われるようになる。これにより、更新処理の二重実行を防止できる。

PRGパターンは内部設計で考慮する事項のため、外部設計ではPRGの一連の画面遷移を1つの画面遷移として扱えばよい。

* クライアントサイド：JavaScriptによるボタン制御

ボタンを押下した際、JavaScriptにより画面を制御することで再度ボタンを押下されないようにする。AP基盤の共通機能として設計・実装するため、個別の処理で設計する必要はない。業務要件により標準の動作以外の設計が必要な場合は、画面仕様に記載すること。

### 不正画面遷移チェック

不正画面遷移とは、業務処理が一連の流れで実行される前提で設計されたシステムにおいて、この一連の流れを無視した画面遷移のことである。例えば、「更新処理を行う画面遷移を直接呼び出す」、「更新処理後の処理結果画面からブラウザの戻るボタン等で直前の画面に戻り、再度更新処理を実行する」などのパターンがあげられる。

本システムでは不正画面遷移の対策として、TERASOLUNA Server Framework for Java 5.x が提供するトランザクショントークンチェック機能を利用する。トランザクショントークンチェックは内部設計で考慮する事項のため、外部設計では個別の処理で設計する必要はない。

## エラー処理

アプリケーションで発生するエラーにはに示す４種類がある。このうち、外部設計において業務開発者が意識すべきエラーは入力エラーと業務エラーである。

表 12　エラーの種類一覧

|  |  |
| --- | --- |
| **エラーの種類** | **概要** |
| 入力エラー | 入力チェックのエラー。詳細は「」で説明した通り。 |
| 業務エラー | 業務ロジックチェック（入力チェックではチェックできない、外部のデータと突き合わせる必要があるチェック）のエラー。発生後は原則としてユーザの画面操作によりシステムの正常ルートへの復帰が可能。  （例）  ・商品を注文する際の在庫切れエラー  ・データの同時更新に起因して発生する排他制御エラー  ・データの同時登録に起因して発生する一意制約違反エラー |
| システムエラー | 業務設計の想定範囲外の異常事態が発生し、以後の処理には復帰できない致命的なエラー。  （例）  ・DB接続エラーや外部システムとの連携エラー  ・Java仮想マシンのメモリ不足エラー |
| 不正リクエストエラー | クライアントからの不正リクエストにより発生するエラー。  （例）  ・POSTメソッドのみ許容しているURLに対して、GETメソッドでアクセスした場合 |

入力チェックのチェック内容は画面仕様に記述する。入力エラー発生時はその後の業務ロジックを実行することなく入力画面に戻り、エラーメッセージを画面に表示する。単項目チェックのエラーメッセージはチェックルールごとに決まっているため、個別の処理で設計する必要はない。相関項目チェックのエラーメッセージは個別の処理ごとに設計する必要がある。

業務ロジックチェックのチェック内容はイベント仕様に記述する。業務エラー発生時は、トランザクションをロールバックし、ユースケース内の画面や共通エラー画面に遷移し、エラーメッセージを画面に表示する。エラーメッセージは個別の処理ごとに設計する必要がある。

## 国際化

国際化対応とは、ユーザメッセージやGUIコンポーネントのラベル名、テキスト要素等をユーザのlocaleに応じた言語で表示することである。

本システムでは、利用者が確定しており、日本語にのみ対応すればよいことが決まっているため、開発コストを考慮し、国際化の対応は実施しないこととする。

## 認証・認可

本システムではユーザID（お客様番号）とパスワードによるユーザ認証処理を行う。ログイン中のユーザについては、そのユーザ情報をシステムで共通的に保持するため、各業務ロジックの中でアクセスすることができる。ユーザ情報を利用したイベント処理を記述する際は、イベント仕様の入力値欄で”ログインユーザ情報”と記述すること。記述例は後述する「設計書の記述要領」を参照。

本システムではログインの有無に応じた認可処理を行う。認可処理には①ユースケース単位でのアクセス制御と②画面、イベント単位での処理分岐がある。①についてはAP基盤の共通機能として設計・実装するため、業務開発者は②について考慮して外部設計を行うこと。

## ファイルアップロード、ダウンロード

本システムでは該当する機能要件が無いため省略。

## 帳票

本システムでは該当する機能要件が無いため省略。

# 設計書の記述要領

## メッセージ一覧

### 記述単位

システムで1部作成する。

### 記述項目

記述項目の一覧をに示す。

表 13 「メッセージ一覧」の記述項目一覧

| ****記述項目**** | ****記述説明**** |
| --- | --- |
| No. | 連番を記述する。 |
| メッセージID | 命名付与基準に従って決定したメッセージのID を記述する。 |
| 出力レベル | メッセージの出力レベルを記述する。 |
| メッセージ | 出力するメッセージを記述する。 |
| 備考 | 特記すべき備考があれば記述する。 |

## 画面一覧

### 記述単位

システムで1部作成する。

### 記述項目

記述項目の一覧をに示す。

表 14 「画面一覧」の記述項目一覧

| ****記述項目**** | ****記述説明**** |
| --- | --- |
| No. | 連番を記述する。 |
| ユースケースグループ名 | 画面が利用されるユースケースグループ名を記述する。 |
| ユースケース名 | 画面が利用されるユースケース名を記述する。 |
| 画面ID | 画面IDを記述する。 |
| 画面名 | 画面名を記述する。 |
| 説明 | 画面の説明を概要レベルで記述する。 |
| 備考 | 特記すべき備考があれば記述する。 |

## 画面遷移図

### 記述単位

基本的にはユースケース単位に作成する。ユースケースごとでは小さすぎる場合は、ユースケースグループごとに作成してもよい。

### 記述項目

記述項目の一覧をに示す。

表 15　「画面遷移図」の記述項目一覧

| ****記述項目**** | ****記述説明**** |
| --- | --- |
| 画面ID/画面名 | 画面遷移図内に画面IDと画面名を記述する。 |
| 画面遷移方向 | 画面間に矢印で記述する。 |
| 画面遷移条件 | 画面遷移するためのイベント（ボタン押下など）を記述する。 |

※UMLのアクティビティ図で作成し、Excel形式の設計書に貼付する。

### 記述方法詳細

#### 画面遷移とイベントの記述方法

画面はアクション要素を使用して作成し、画面IDと画面名を記述する。画面IDはステレオタイプを追加して記述する。画面遷移は画面間を制御フロー要素で連結することで表現し、イベントは制御フローのアクション欄を使用して”(イベントID)イベント名”の形式で記述する。記述例をに示す。

正常系遷移：制御フロー要素

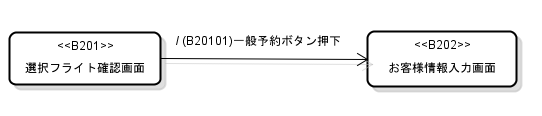
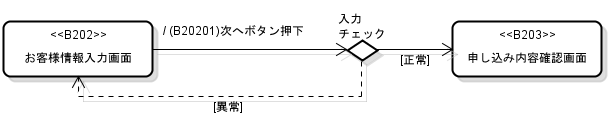


図 3　画面遷移とイベントの記述例

#### 処理結果により遷移先が異なる場合の記述方法

同一のイベントで処理結果や条件により遷移先が異なる場合は、デシジョンノード要素を使用して分岐を表現する。異常系の遷移は破線で記述する。記述例をに示す。

分岐：デシジョンノード要素



[図 4 処理結果により遷移先が異なる場合の記述例](\\\\10.252.20.80\\fy2013\\03_CASE\\03_TERASOLUNA_FWの開発\\00_管理\\10_企画・戦略\\H25\\20130816 NTT統合FW\\60 CF2\\work\\EPF\\epf-composer\\terasoluna.open.core\\guidances\\guidelines\\resources\\different_transition.emf)

#### ユースケース間の画面遷移の記述方法

他ユースケースとのインターフェースとなる画面遷移を記述する場合、フォークノード要素を使用してユースケース境界が識別できるようにする。また、ユースケース外画面は色を変えて（背景：灰色）識別できるようにする。ユースケース境界とユースケース外画面間は破線で接続する。記述例をに示す。

ユースケース外画面

ユースケース境界：フォークノード要素

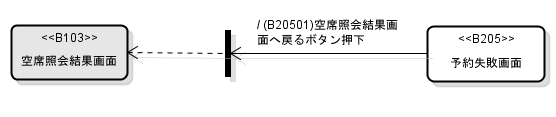


図 5　ユースケース間の画面遷移の記述例

## 画面仕様

### 記述単位

画面単位に作成する。1つの画面仕様は「レイアウト」「画面項目」「画面状態」の３シートで構成される。

### レイアウトシート

### 記述項目

記述項目の一覧をに示す。

表 16　｢レイアウト｣シートの記述項目一覧

| 記述項目 | 記述説明 |
| --- | --- |
| 画面名 | 画面名を記述する。 |
| 画面ID | 画面IDを記述する。 |
| 画面レイアウト | UI規約に基づいて画面のレイアウトを決定し、イメージを貼付する。 |

### 記述方法詳細

#### 画面レイアウトはHTMLで作成する

画面レイアウトはHTMLで作成し、スクリーンキャプチャしたものを貼り付ける。

#### 項目ごとに番号を付与する

画面の各項目に番号を付与し、画面項目シートの項目番号と対応させる。

#### 共通部品を明示する

画面キャプチャに共通部品が含まれる場合は、枠で囲うなどして、共通部品を明示する。

#### メッセージ表示領域を明示する

画面にメッセージを表示する場合、画面キャプチャ中の表示箇所を枠で囲うなどして、メッセージ表示領域を明示する。

### 画面項目シート

### 記述項目

記述項目の一覧をに示す。

表 17　｢画面項目｣シートの記述項目一覧

| 記述項目 | | 記述説明 |
| --- | --- | --- |
| 画面名 | | 画面名を記述する。 |
| 画面ID | | 画面IDを記述する。 |
| No. | | 連番を記述する。 |
| 論理項目名 | | 項目の論理名を記述する。  項目をグループ化する場合は、論理項目名を階層化して記入する。 |
| 論理項目種別 | | 項目の種別（ラベル、テキストボックス、リストボックスなど）を記述する。  指定できる種別の種類は後述の一覧を参照。 |
| 入力 | | 入力項目かどうかを記述する。  ○：入力項目  △：状況に応じて入力項目となり得る  －：入力項目ではない  「△」の場合には「その他入力条件」に入力可能となる条件を記述する。 |
| 出力 | | 出力項目かどうかを記述する。  ○：出力項目  △：状況に応じて出力項目となり得る  　　　（出力する際の条件を「表示条件など」  　　　欄、または「初期値」欄に記載する）  - ：出力項目ではない |
| 必須 | | 入力が必須かどうかを記述する。  ○：必須項目  △：状況に応じて必須項目となり得る  　　　（必須となる条件を  　　　「その他入力条件」欄に記載する）  - ：必須項目ではない |
| 入力桁数 | 最小 | 項目の最小入力桁数を記述する。 |
| 最大 | 項目の最大入力桁数を記述する。 |
| 文字種別 | | 項目に入力される文字の文字種別を記述する。  指定できる種別の種類は後述の一覧を参照。 |
| その他入力条件 | | 単項目チェックのうち、必須、桁数範囲、文字種別以外のチェックルール（クレジットカード番号形式、Emailアドレス形式、URL形式、未来・過去日付、最小・最大値）を記述する。  相関項目チェックのルールを記述する。  ※業務ロジックチェック（DBやセッションなどのコンテキストを用いて行う業務に関わるチェック）はこの欄に記述せずイベント仕様に記述する。 |
| 初期値 | | 項目の初期値の有無と設定値を記述する。  初期値として明示的に空白を示す必要がある場合は""（ダブルクォーテーション２つ）を記述する。 |
| 表示条件など | | 項目の表示条件などがあれば記述する。  金額や日付などの表示フォーマットを指定する際は、具体的に「”###,###,###円”の形式で出力」「”yyyy年MM月dd日”の形式で出力」のように記述する。  リストボックス、コンボボックスの選択候補についてはこの欄に記述する。 |
| 発行イベントID | | 項目から発生するイベントを抽出し、インベントIDを記述する。  イベントのトリガーが「ボタン押下」以外の場合、その情報をイベントIDの後にカッコ付きで記述する。(例)　B20101(フォーカス時) |
| チェックタイミング | | 入力チェックを行うタイミングを記述する。 |

論理項目種別に指定できる種別の種類一覧をに示す。

表 18　論理項目種別の種類一覧

| **カテゴリ** | **論理項目種別** |
| --- | --- |
| 入力・選択 | テキストボックス |
| パスワードボックス |
| リストボックス |
| ラジオボタン |
| チェックボックス |
| ファイルフィールド |
| テキストエリア |
| 表示 | ラベル |
| リンク |
| イメージリンク |
| ページネーションリンク |
| ボタン | ボタン |
| リセットボタン |
| 画像ボタン |

文字種別に指定できる種別の種類一覧をに示す。

表 19　文字種別の種類一覧

| **文字種別** | **備考** |
| --- | --- |
| 半角文字 | 半角文字の集合 |
| 半角英数字 | 半角英字文字、半角数字文字の集合 |
| 半角英字 | 半角英字文字の集合 |
| 半角数字 | 半角数字文字の集合 |
| 半角カタカナ | 半角カタカナ文字の集合 |
| 全角文字 | 全角文字の集合 |
| 全角英数字 | 全角英字文字、全角数字文字の集合 |
| 全角英字 | 全角英字文字の集合 |
| 全角数字 | 全角数字文字の集合 |
| 全角カタカナ | 全角カタカナ文字の集合 |
| 全角+半角文字 | 全角文字、半角文字の集合 |

### 記述方法詳細

#### 一覧をページ分割して表示する場合の記述方法

一覧をページ分割して表示する方法（ページネーション）を利用する場合、「表示条件など」欄に1ページ当たりの表示件数とソート条件を記述する。

例)　1ページ当たりの表示件数：10件、ソート条件：ユーザIDで昇順

### 画面状態シート

｢画面状態｣は画面の状態が変わる場合に定義する。例えば、ユーザ権限やワークフローによって画面の表示状態が変化する場合などが該当する。ただし、状態変化が特定の画面項目に閉じている場合（チェックボックスにチェックを入れると特定のフィールドが入力可能になる、など）は、画面項目の「表示条件など」欄へのみ記述し、画面状態シート欄への記載を省略してもよい。

### 記述項目

記述項目の一覧をに示す。

表 20 「画面状態」シートの記述項目一覧

| ****記述項目**** | ****記述説明**** |
| --- | --- |
| 画面名 | 画面名を記述する。 |
| 画面ID | 画面IDを記述する。 |
| No. | 連番を記述する。 |
| 論理項目名 | 画面項目の論理項目名を記述する。  項目をグループ化する場合は、論理項目名を階層化して記入する。 |
| 状態 | 発生しうる状態を洗い出し、それぞれの状態における項目ごとの表示状態を記述する。初期値など決まった値を表示する際は、その値を記述する。 |

### 記述方法詳細

#### 表示状態の記述方法

画面項目の表示状態はを参考に決定する。以下のルールに当てはまらない場合は、わかりやすい表示状態名を付与すること。

表 21 表示状態の記述方法

| **表示状態名** | **対象画面項目種別** | **説明** |
| --- | --- | --- |
| 表示/非表示 | すべての画面項目 | ある条件に一致した場合のみ、表示/非表示するような項目。  例）画面の初期表示時のエラーメッセージ  　　検索画面の初期表示時の検索結果（非表示） |
| 入力値 | テキストボックスなどの入力用コントロール | 前画面などでユーザが入力/選択した値を表示する場合。  例）検索結果画面で、前画面で入力された検索条件を表示するテキストボックス |
| 有効/無効 | ボタンなどのコントロール一般 | コントロールを入力、選択不可状態にする場合。  例）登録権限のないアクターがアクセスした場合の登録ボタン（無効） |
| 読み取り専用 | テキストボックスなどの入力用コントロール | 項目が読み取り専用である場合。  例）DBから取得したデータを表示するためのテキストボックス（読み取り専用） |

#### 同一のボタンで遷移先が異なる場合の記述方法

「前ページに戻る」という名前のボタンで、状態によって戻り先が異なるボタンが存在する場合、画面項目としては２つの項目を定義し、画面状態で遷移状態ごとに表示/非表示を定義すること。記述例をに示す。

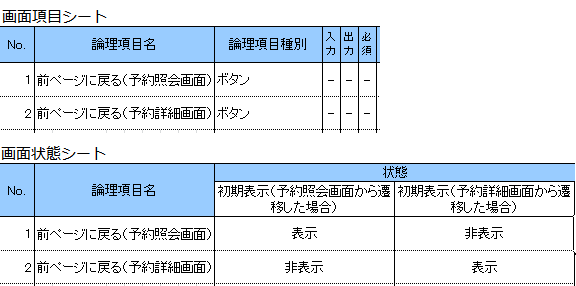


図 6　同一のボタンで遷移先が異なる場合の記述例

## イベント一覧

### 記述単位

システムで1部作成する。

### 記述項目

記述項目の一覧をに示す。

表 22 「イベント一覧」の記述項目一覧

| ****記述項目**** | ****記述説明**** |
| --- | --- |
| No. | 連番を記述する。 |
| ユースケースグループ名 | イベントが発生するユースケースグループ名を記述する。 |
| ユースケース名 | イベントが発生するユースケース名を記述する。 |
| 画面ID | イベントが発生する画面IDを記述する。 |
| イベントID | イベントIDを記述する。 |
| 発出元イベントID | ユースケースをまたがるイベントの場合、その発出元のイベントIDを記述する。 |
| イベント名 | イベント名を記述する。 |
| イベント区分 | イベントが発生する場所の区分を記述する。  S:サーバイベント  C:クライアントイベント |
| イベント仕様 | イベント仕様の作成有無を記述する。  ○：作成する  ―：作成しない |
| 概要 | イベントの説明を概要レベルで記述する。 |

### 記述方法詳細

#### ユースケースをまたがったイベントの場合の記述方法

ユースケースをまたがったイベント(異なるユースケースの画面を表示するイベントなど)の場合、それぞれ個別にイベントIDを設定する。

例）

画面<<B103>> → イベントB10301 → | → イベントB20106 → 画面<<201>>

画面仕様書にはボタン押下時の発生イベントを記載すること。

#### イベント仕様の作成が省略可能なケース

下記の条件にあてはまる場合はイベント仕様の作成を省略してもよい。

1. ユースケースをまたがるイベントの内、発出元となるイベント(概要欄に発行イベントのイベントIDを記載すること)
2. 簡易なクライアントイベント
3. 戻るボタンイベント等、画面表示のみのイベント
4. 他のイベントと処理内容が同様のため省略可能なイベント(概要欄に同様であるイベントのイベントIDを記載すること)

## イベント仕様

### 記述単位

ユースケース単位に作成する。ユースケースに含まれるイベントごとにシートを作成する。

### 記述項目

記述項目の一覧をに示す。

表 23 「イベント仕様」の記述項目一覧

| ****記述項目**** | | | ****記述説明**** |
| --- | --- | --- | --- |
| イベントID | | | イベントIDを記述する。 |
| イベント名 | | | イベント名を記述する。 |
| 概要 | | | イベントの概要を記述する。 |
| 入力値 | | | |
|  | No. | | 連番を記述する。 |
| 論理項目名 | | イベント処理に対する入力情報を記述する。導出元が画面なら画面仕様の論理項目名と一致させること。  項目をグループ化する場合は、論理項目名を階層化して記入する。 |
| 導出元 | | 入力値の導出元を記述する。 |
| 備考 | | 特記すべき備考があれば記述する。 |
| 処理ロジック | | | |
|  | | No. | 連番を記述する。 |
| 処理 | イベントの処理内容を記述する。必要に応じて事前条件や異常処理なども記述する。 |
| 出力先 | イベントの処理の結果、画面遷移が発生する場合、遷移先画面の画面名と画面IDを記述する。入力エラーや業務エラーが契機となり遷移する場合は、その条件とエラーメッセージを併せて記述する。 |
| 出力値 | | | |
|  | | No. | 連番を記述する。 |
| 論理項目名 | イベントの処理の結果、画面に出力される情報を記述する。画面仕様の論理項目名と一致させること。  項目をグループ化する場合は、論理項目名を階層化して記入する。 |
| 備考 | 特記すべき備考があれば記述する。 |

### 記述方法詳細

#### 入出力値がリスト項目の場合の記述方法

入出力値がリスト項目の場合は、論理項目名を階層化した上で上位階層の項目名を「xxxx一覧」のように記述する。また、リストの最大件数が明確になっている場合は「xxxx一覧（最大X件）」のように記述する。記述例をに示す。

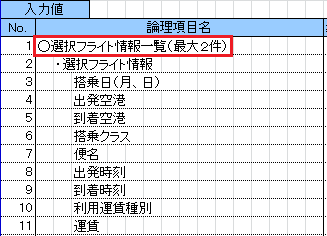


図 7　入出力値がリスト項目の場合の記述例

#### 入力値の導出元の記述方法

入力値は論理項目ごとに導出元を記述する。項目がグループ化されている場合は上位階層の項目にのみ対応する導出元を記述すればよい。導出元の例としてはイベントが発生した画面や、ログインユーザ情報が挙げられる。記述例をに示す。

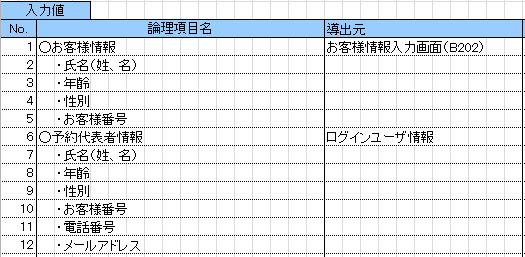


図 8　入力値の導出元の記述例

#### 処理ロジックの記述方法

##### 制御構造（分岐、繰り返しなど）

処理の分岐がある場合は、分岐条件と分岐後の処理を記述する。処理の繰り返しがある場合は、繰り返しの回数や終了条件と繰り返す処理を記述する。記述例をリスト1に示す。

リスト 1　制御構造の書き方の例

|  |
| --- |
| **"運賃種別"がレディース割の場合、入力された"お客様情報"が全て女性であるかチェックする。** |

##### 入力チェック（単項目チェック、相関項目チェック）

単項目チェックと相関項目チェックは画面仕様にチェック内容が記述されているため、イベント仕様の処理ロジックの中ではチェックを実行する旨のみ記述する。記述例をリスト2に示す。

リスト 2　入力チェックの書き方の例

|  |
| --- |
| **発出元画面の仕様に従い、画面入力情報の単項目/相関チェックを実行する。** |

##### エンティティへのアクセス

処理ロジックの中でDB等のエンティティにアクセスする際は、対象のエンティティ名を明記した上で、CRUD種別ごとに以下の観点で処理内容を記述する。記述例をリスト3に示す。

・C：作成

　　作成するデータの導出元（画面項目、計算値、初期値）

・R：参照

　　参照するエンティティ名、取得件数、レコードの絞り込み条件、ソート条件（複数レコード参照する場合）

・U：更新

　　更新するエンティティ名、データの導出元（画面項目、計算値、初期値）、レコードの絞り込み条件

・D：削除

　　削除するレコードの絞り込み条件

リスト 3　DBアクセスの書き方の例

|  |
| --- |
| **入力された"お客様情報"の"お客様番号"に該当する"ATRSカード会員情報"をDBから1件取得する。** |

##### ビジネスルールの参照

処理ロジックの中でビジネスルールに基づいて演算処理や条件分岐を行う際は、参照するビジネスルール名を記述する。記述例をリスト4に示す。

リスト 4　ビジネスルールを参照する書き方の例

|  |
| --- |
| **"合計金額"を算出する。**  **※"合計金額"の算出方法については要件定義書 ビジネスルール：「合計金額の算出方法について」を参照。** |

1. TERASOLUNA Server Framework for JavaにおけるOSSスタックの詳細はTERASOLUNA Server Framework for Java (5.x)Development Guideline「2.1 TERASOLUNA Server Framework for Java (5.x)のスタック」を参照のこと。 [↑](#footnote-ref-1)
2. 本システムの開発では作業する人数が少数であり、作業内容の統一のためにUI規約を作成する必要がないため、作成対象外としている。 [↑](#footnote-ref-2)
3. アプリケーションのアーキテクチャ策定から業務横断的な共通機能の提供を行うことで、各アプリケーション開発者を支援するチーム。 [↑](#footnote-ref-3)
4. 画面遷移図は原則ユースケース単位に作成するが、ユースケースごとでは小さすぎる場合は、ユースケースグループごとに作成してもよい。 [↑](#footnote-ref-4)
5. Emailアドレスのフォーマットを定めた標準仕様<http://www.rfc-editor.org/rfc/rfc2822.txt>

   　標準仕様と日本の携帯電話会社が許可しているアドレス形式では差があるため、必要に応じて対応するルールを独自に定めること。 [↑](#footnote-ref-5)
6. URLのフォーマットを定めた標準仕様<http://www.ietf.org/rfc/rfc2396.txt> [↑](#footnote-ref-6)