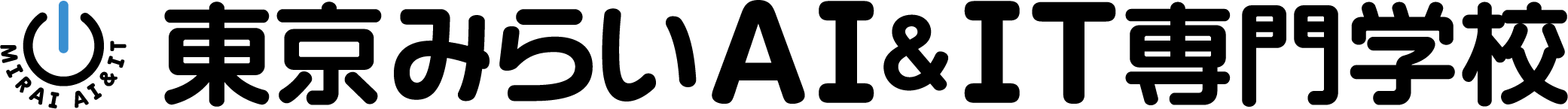
関係社外秘



| 詳細設計書 |
| --- |



目次

[１ プログラム構成 2](#_heading=h.30j0zll)

[１-１ プログラム全体構成 2](#_heading=h.1fob9te)

[１-２ パッケージ一覧 3](#_heading=h.3znysh7)

[１-3 テンプレートファイル構成](#_heading=h.tyjcwt) 4

[１-4 URL一覧](#_heading=h.3dy6vkm) 6

[２ ユーザインターフェース設計](#_heading=h.1t3h5sf) 7

[２-１ 共通部品とレイアウトパターン](#_heading=h.4d34og8) 7

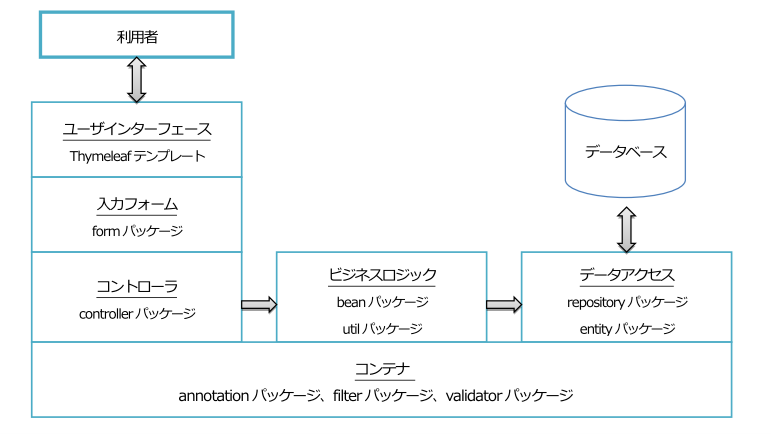
[２-２ 入力値チェック](#_heading=h.2s8eyo1) 11

[３ プログラム構造設計 1](#_heading=h.17dp8vu)2

[３-１ 概要 1](#_heading=h.3rdcrjn)2

[３-２ アクセス制御の方式 1](#_heading=h.26in1rg)2

4 [技術的な制約・既知の問題点 1](#_heading=h.2xcytpi)3

1. プログラム構成
   1. プログラム全体構成

(1) ユーザインターフェース

・Thymeleafのテンプレートを利用する。

・コントローラとの情報の受渡は、Modelオブジェクト、Sessionスコープを利用する。

・画面間の情報の受渡は、必要に応じてformタグのhidden属性を利用する。

(2)入力フォーム

・入力画面に合わせて、Formクラスを用意する。

・Formクラス内で入力値チェックのアノテーションを指定する。

(3)コントローラ

・一般会員用と管理者(運用管理者・システム管理者)用のコントローラを用意する。

・＠RequestMappingを用いて、利用者からのリクエストに対応するメソッドを用意する。

・ビジネスロジック、データアクセスクラスと連携し、各機能を実装する。

(4)ビジネスロジック

・情報保持をするためのBeansクラスを用意する。

・データの整形処理、データチェック処理、計算処理、定数定義用のクラスを用意する。

(5)データアクセス

・データベース操作のためのクラスを用意する。

(6)コンテナ

・システム全体で必要となるログイン認証状態の確認や独自の入力チェック処理を用意する。

* 1. パッケージ一覧

Spring Frameworkを利用し、下記のパッケージ構成とする。

| パッケージ名(アルファベット順) | 利用用途 |
| --- | --- |
| jp.co.aico.bean | JavaBeans |
| jp.co.aico.controller.registration | 登録機能のコントローラ |
| jp.co.aico.controller.login | ログイン機能のコントローラ |
| jp.co.aico.controller.help | ヘルプ機能のコントローラ |
| jp.co.aico.controller.com | 会話機能のコントローラ |
| jp.co.aico.controller.quiz | クイズ機能のコントローラ |
| jp.co.aico.controller.management | 管理機能のコントローラ |
| jp.co.aico.controller.add | 追加機能のコントローラ |
| jp.co.aico.controller.user | 会員管理機能のコントローラ |
| jp.co.aico.entity | データベース操作 |
| jp.co.aico.form | 入力フォームの処理 |
| jp.co.aico.repository | データベース操作 |
| jp.co.aico.util | 共通処理、ビジネスロジック |
| jp.co.aico.validator | 入力値のチェック処理 |
| jp.co.aico.filter | セッション情報チェック、割り込み処理 |

* 1. テンプレートファイル構成

・ユーザインターフェースは、Thymeleafのテンプレートを利用する

・テンプレートファイルは、/resources/template配下に置く

・各画面は、いずれかのレイアウトパターン(詳細は、後述)を利用する

・/resources/template配下のパス名・ファイル名を下記に示す

・背景色がある箇所はレイアウト機能を使用する場合にのみ作成予定

**・ヘッダからサイドバーは、springのレイアウト機能を使用する。(大谷先生に承諾済み)**

**実装できるかどうかは、製造しながら決めていく。**

　 ・(使い方に関しては、メンターとユーザーで分けていたほうが見やすいと思ったので

　　使い方の画面は分けています。)

画面名

| 画面名 | テンプレート内にあるフォルダへのパス | ファイル名 |
| --- | --- | --- |
| ヘッダ | /common | header.html |
| ナビゲーションバー | /common | menu.html |
| フッタ | /common | footer.html |
| サイドバー | /common | sidebar.html |
| レイアウトパターン1 | /common | layout1.html |
| レイアウトパターン2 | /common | layout2.html |
| レイアウトパターン3 | /common | layout3.html |
| ログイン画面 | /login | login.html |
| 管理者用メニュー画面 | /admin | admin\_menu.html |
| メンター用メニュー画面 | /mentor | mentor\_menu.html |
| トップ画面 | /top | top.html |
| ログイン後トップ画面 | /top | dotop.html |
| 設定画面 | /set | set.html |
| 会員詳細情報表示画面 | /user | user\_info.html |
| 会員情報一覧表示画面(管理者用) | /user | user\_list.html |
| 会員登録入力画面 | /user/regist | user\_regist\_input.html |
| 会員登録確認画面 | /user/regist | user\_regist\_check.html |
| 会員登録完了画面 | /user/regist | user\_regist\_complete.html |
| 会員変更入力画面 | /user/update | user\_update\_input.html |
| 会員変更確認画面 | /user/update | user\_update\_check.html |
| 会員変更完了画面 | /user/update | user\_update\_complete.html |
| 会員削除確認画面 | /user/delete | user\_delete\_check.html |
| 会員削除完了画面 | /user/delete | user\_delete\_complete.html |
| Q&A画面 | /help | help\_question.html |
| 使い方画面(一般会員) | /help/guide | help\_guide\_user.html |
| 使い方画面(メンター) | /help/guide | help\_guide\_mentor.html |
| 問い合わせ入力画面 | /help/question | help\_question\_input.html |
| 問い合わせ確認画面 | /help/question | help\_question\_check.html |
| 問い合わせ完了画面 | /help/question | help\_question\_complete |
| 問い合わせ閲覧画面 | /help/question/answer | help\_question\_answer\_view.html |
| クイズ一覧画面 | /quiz | quiz\_list.html |
| クイズ出題、回答画面 | /quiz | quiz\_question.html |
| クイズ解説画面 | /quiz | quiz\_explain.html |
| クイズ履歴閲覧画面 | /quiz/log | quiz\_log\_view.html |
| zoomリンク掲載画面 | /zoom | zoom\_transition.html |
| 日程予約入力画面 | /calendar | calendar\_view.html |
| 日程予約確認画面 | /calendar | calendar\_check.html |
| 日程予約完了画面 | /calendar | calendar\_complete.html |
| チャット閲覧、入力画面 | /chat | chat\_view.html |
| チャット確認画面 | /chat | chat\_check.html |
| チャット完了画面 | /chat | chat\_complete.html |

※お問い合わせ入力/確認/完了画面はGoogle Formを使用する

* 1. URL一覧　　**(※省略)**

・http://localhost:55000/shared\_shop/に続くパス名の一覧を下記に示す

・テンプレートのformタグのaction属性やaタグのhref属性に指定する

・指定されたパスと処理を対応づけるために＠RequestMappingを利用し、コントローラクラスのメソッドとパス名を紐づける

・該当するボタンやリンクに付与されるURLの形式をあらわしている

「xxxx」：固定値、{xxxx}：可変値

| 画面ID | 画面名・部品名 | テンプレート内指定箇所 | パス名 | メソッド |
| --- | --- | --- | --- | --- |
| － | ヘッダ | 「ログイン」リンク | /login | GET |
| － | ヘッダ | 「新規会員登録」リンク | /user/regist/input | GET |
| － | ヘッダ(一般会員) | {会員氏名} リンク | /user/detail | GET |
| － | ヘッダ | 「ログアウト」リンク | /logout | GET |
| － | ヘッダ (運用管理者・システム管理者) | {会員氏名} リンク | /user/detail/admin/{id} | GET |
| － | ナビゲーションバー | 「トップ」リンク | / | GET |
| － | ナビゲーションバー | 「新着一覧」リンク | /item/list/1 | GET |
| － | ナビゲーションバー | 「買い物かご」リンク | /basket/list | GET |
| － | ナビゲーションバー | 「注文一覧」リンク | /order/list | GET |
| － | サイドバー (非会員・一般会員) | カテゴリ別検索「検索」ボタン | /item/list/category/1?categoryId={カテゴリID} | GET |
| － | サイドバー(運用管理者) | 「会員一覧表示」リンク | /user/list | GET |
| － | サイドバー(運用管理者) | 「注文一覧表示」リンク | /order/list/admin | GET |
| － | サイドバー(運用管理者) | 「商品一覧表示」リンク | /item/list/admin | GET |
| 1-1-1-1 | ログイン画面 | 「ログイン」ボタン | /login | POST |
| 1-3-1-3 1-3-1-4 | 管理者用メニュー画面 | 「会員一覧表示」リンク | /user/list | GET |
| 1-3-1-3 | 管理者用メニュー画面 | 「注文一覧表示」リンク | /order/list/admin | GET |
| 1-3-1-3 | 管理者用メニュー画面 | 「商品一覧表示」リンク | /item/list/admin | GET |
| 1-3-1-3 | 管理者用メニュー画面 | 「カテゴリ一覧表示」リンク | /category/list | GET |
| 2-1-1-1 2-1-1-2 | トップ画面 | {商品画像}リンク | /item/detail/{id} | GET |
| 2-2-1-1  2-2-1-2 | 商品一覧画面 | 「新着順」「売れ筋順」リンク | /item/list/{sortType} | GET |

1. ユーザインターフェース設計
   1. 共通部品とレイアウトパターン

Thymeleafのレイアウト機能を利用し、別ファイルで作成した共通部品を埋め込む形式とする。

(1) 共通部品

ヘッダ ファイル名：/common/header.html

ナビゲーションバー ファイル名：/common/menu.html

サイドバー ファイル名：/common/sidebar.html

フッタ ファイル名：/common/footer.html

(2)レイアウト

レイアウトは、３つのパターンを用いる。

<レイアウトパターン１>

構成要素：ヘッダ、メインコンテンツ、フッタ、サイドバーの4ブロック構成

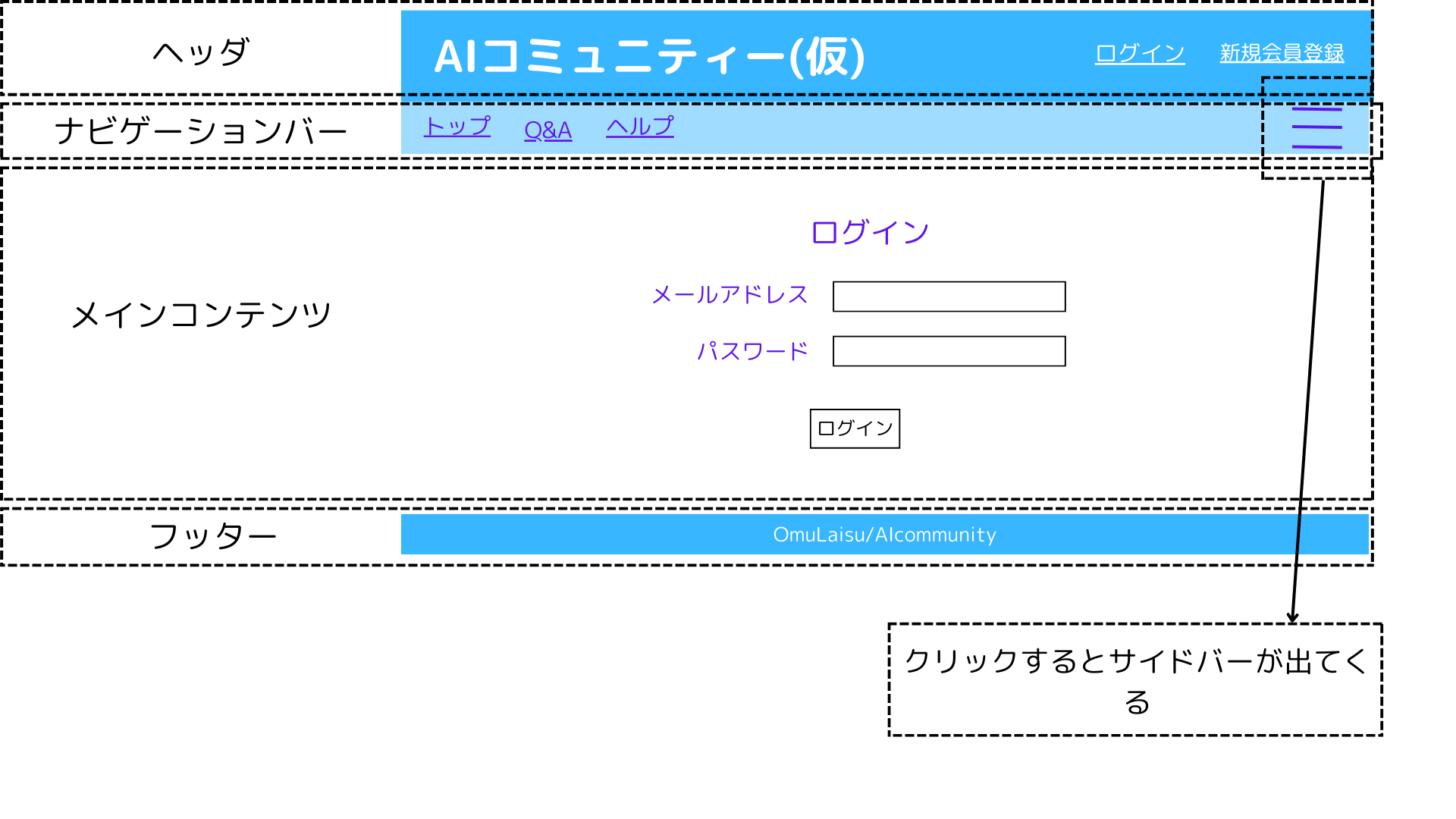
レイアウトファイル名：/common/layout1.html

使用画面 ：

ログイン画面

管理者用メニュー画面

エラー画面

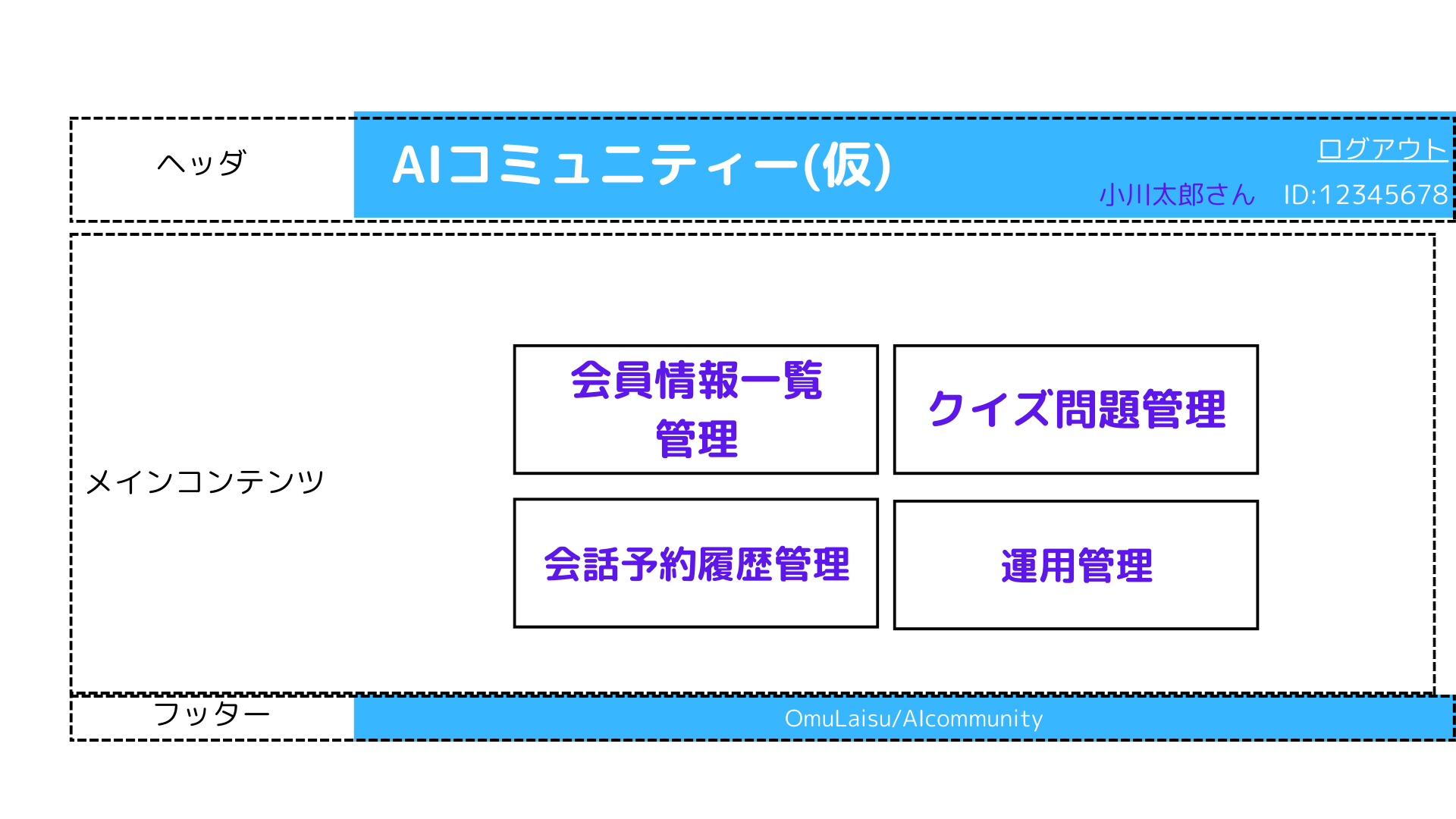


<レイアウトパターン2>

構成要素： ヘッダ、メインコンテンツ、フッタ(ホーム画面でクリックできる箇所が大きく分けて今のところ4つなのでナビゲーションバーは入れません)

レイアウトファイル名：/common/layout2.html

使用画面 ： 運用管理者、システム管理者用画面



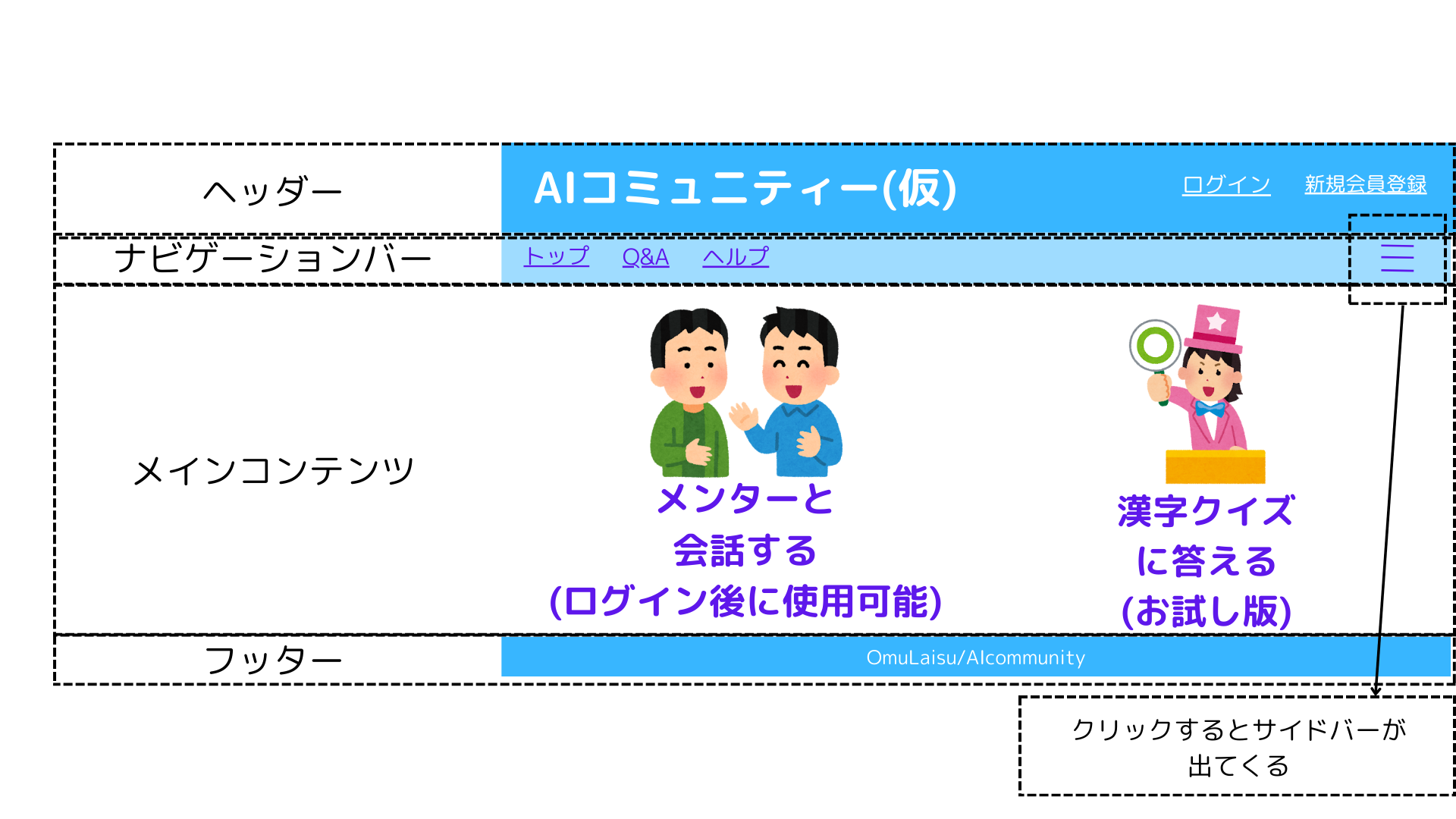
ヘッダ、メインコンテンツ、フッタ

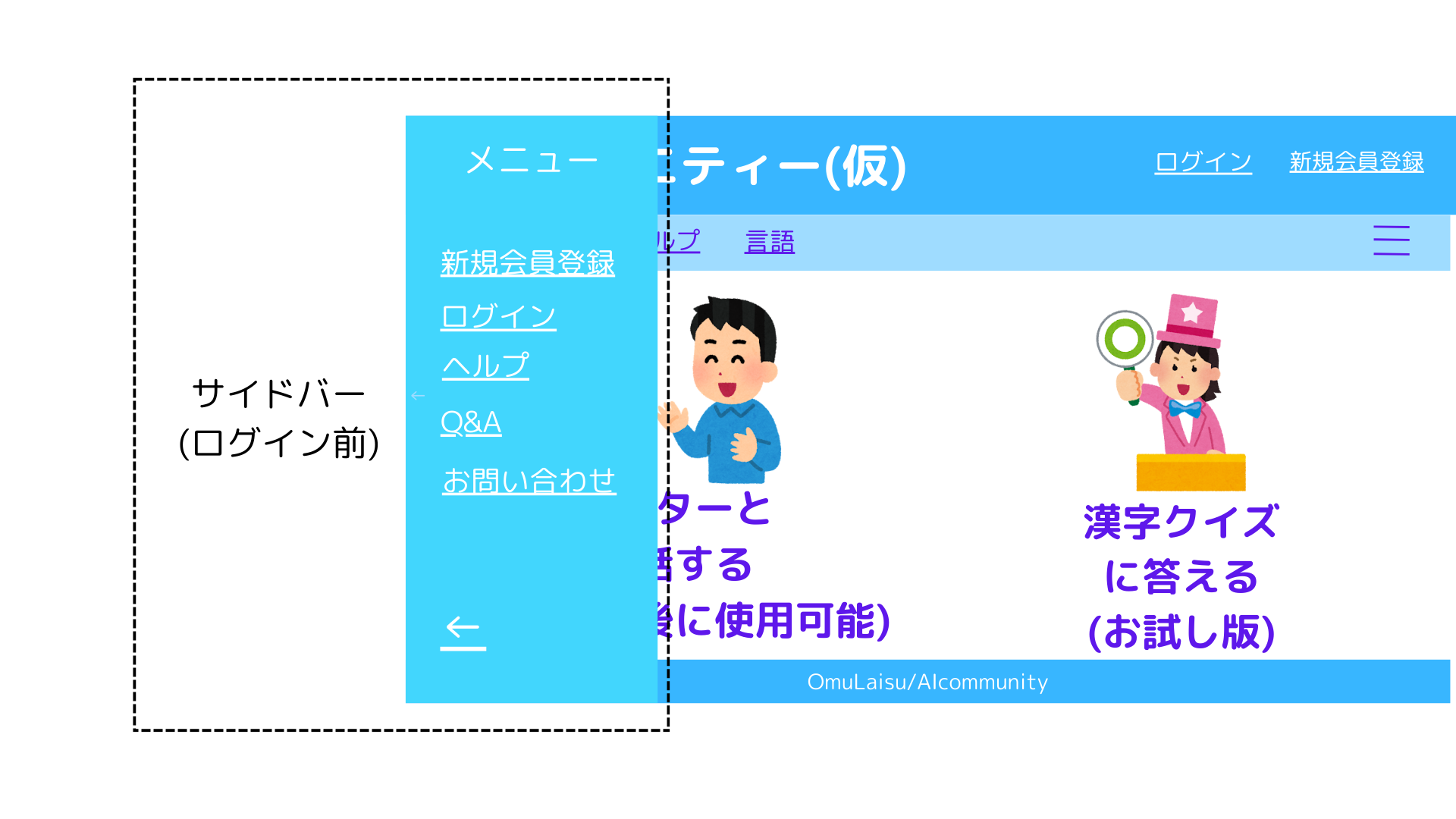
<レイアウトパターン３>

構成要素：ヘッダ、ナビゲーションバー、メインコンテンツ、フッタ、サイドバー

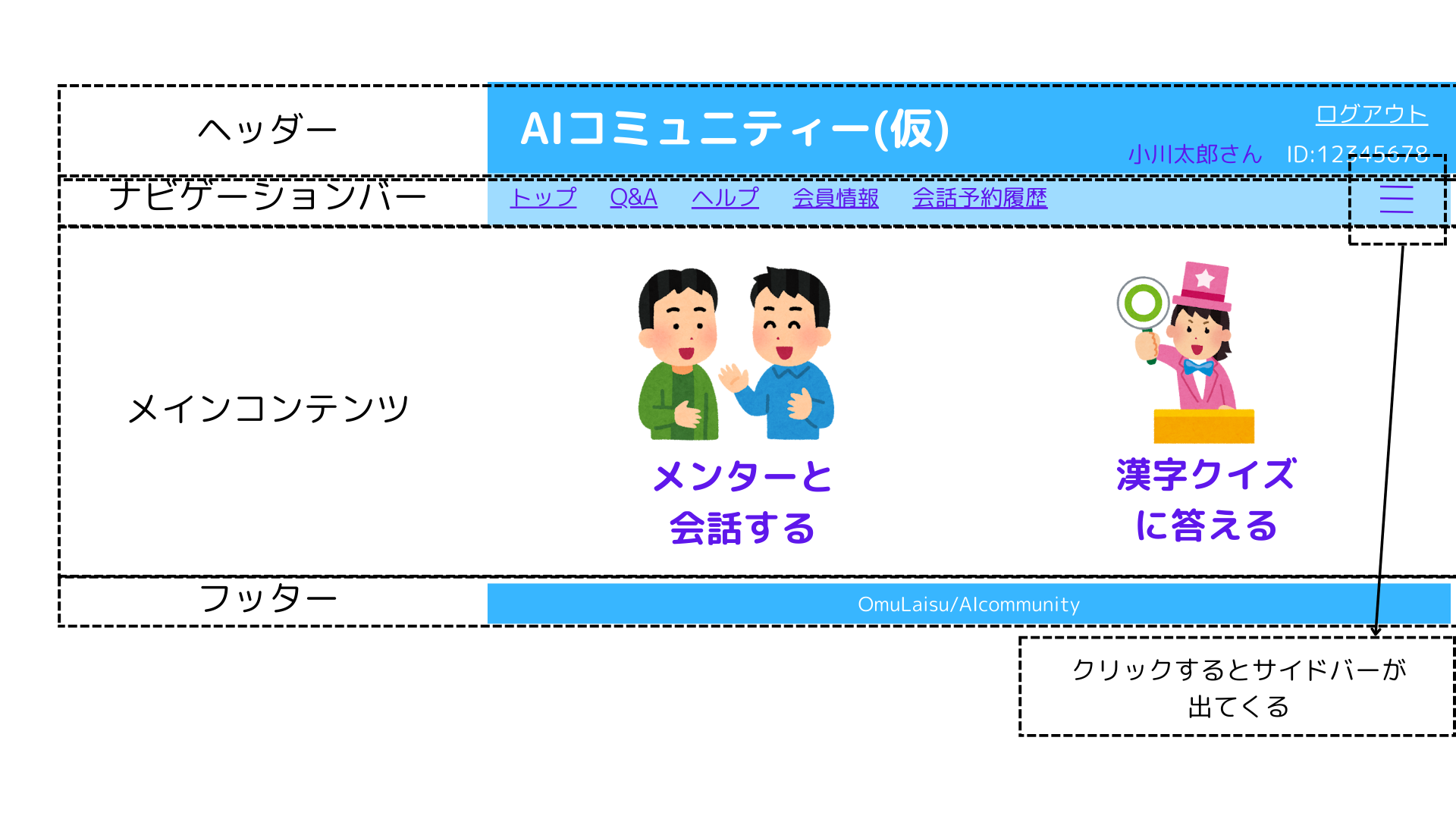
レイアウトファイル名：/common/layout3.html

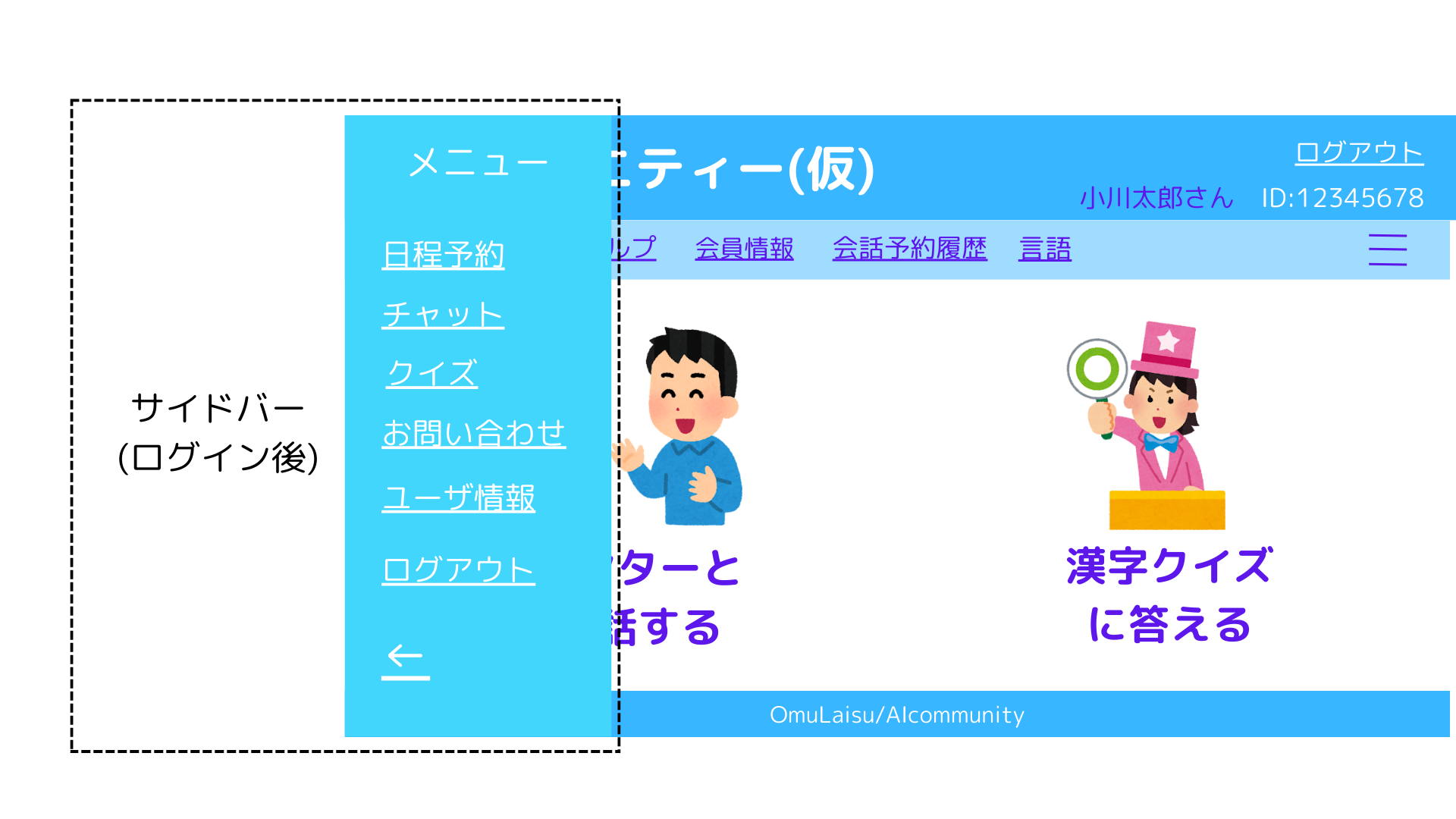
使用画面 ： 非会員、一般会員用画面

非会員用画面



　　　 会員用画面





* 1. 入力値チェック

・入力値の条件と入力チェック用のアノテーションを下記に記載する。

・アノテーションは、javax.validation.constraintsパッケージのアノテーションを利用する。

・以下3点については、独自アノテーションを利用する

－ログイン認証 @LoginCheck

－メールアドレス重複チェック ＠EmailCheck

－カテゴリ名重複チェック @CategoryCheck

(1) ログイン情報

入力画面：1-1-1-1 ログイン画面

| 項目名 | 形式 | 最小値／ 最小文字数 | 最大値／  最大文字数 | 利用アノテーション |
| --- | --- | --- | --- | --- |
| メールアドレス | メールアドレス形式 （RFC2822準拠） | 1 | 129  ＠前後64文字 | @NotBlank  @Email |
| パスワード | 半角英数字 | 8 | 16 | @NotBlank  @Size @Pattern |

(2)会員情報

入力画面：6-3-1-1、6-3-1-3、6-3-1-4 会員登録入力画面

6-4-1-2、6-4-1-3、6-4-1-4 会員変更入力画面

| 項目名 | 形式 | 最小値／ 最小文字数 | 最大値／  最大文字数 | 利用アノテーション |
| --- | --- | --- | --- | --- |
| メールアドレス | メールアドレス形式 （RFC2822準拠） | 1 | 129  ＠前後64文字 | @NotBlank  @Email |
| パスワード | 半角英数字 | 8 | 16 | @NotBlank  @Size  @Pattern |
| 氏名 | 半角文字、全角文字 | 1 | 30 | @NotBlank  @Size |
| 郵便番号 | 半角数字  ハイフン「-」なし | 7 | 8 | @NotBlank  @Size  @Pattern |

1. プログラム構造設計
   1. 概要

拡張性、再利用性、セキュリティ侵害に対する堅牢性を考慮したプログラム構造とするために

下記の点を決定した。

・アクセス制御の方式

* 1. アクセス制御の方式

・利用者の権限に応じてアクセスできる機能を制限するために、フィルタを利用する。

・権限の判別は、以下の方法で行う

* SessionAttributeにログイン情報があるか否か
* ログイン情報がない場合、非会員機能及び会員削除完了画面
* ログイン情報がある場合、権限は一般会員、メンター、運用管理者のいずれか

(1) 権限一覧

| 機能名 | | | | **権限 ○：利用可能 ×：利用できない** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 大分類 | | **小分類** | | **非会員**  **(未ログイン)** | **一般会員** | **メンター** | **運用管理者** |
| 1 | 会員登録 | 1 | ログイン | 〇 | - | - | - |
| 2 | ログアウト | － | ○ | ○ | ○ |
| 2 | ヘルプ | 1 | 使い方 | 〇 | ○ | 〇 | - |
| 2 | よくある質問(Q&A) | 〇 | ○ | 〇 | 〇 |
| 3 | お問い合わせ | × | ○ | 〇 | - |
| 4 | 質問(掲示板) | × | ○ | 〇 | - |
| 3 | 会話 | 1 | カレンダー | × | 〇 | 〇 | ○ |
| 2 | メンターとのチャット | × | 〇 | 〇 | ○ |
| 4 | クイズ | 1 | クイズ出題・解答 | 〇 | ○ | × | × |
| 2 | 漢字(日本語)学習 | 〇 | ○ | × | 〇 |
| 3 | 解きなおし | × | ○ | × | 〇 |
| 4 | ビジネスマナー講習 | × | 〇 | × | 〇 |
| 5 | 解説機能 | × | 〇 | × | 〇 |
| 5 | 管理 | 1 | 進捗度 | × | ○※1 | × | ○※2 |
| 2 | パスワードの再設定 | × | ○ | 〇 | ○ |
| 6 | 追加 | 1 | 翻訳 | 〇 | 〇 | ○ | ○ |
| 7 | 会員管理 | 1 | 会員一覧表示 | × | × | 〇※3 | 〇 |
| 2 | 会員詳細表示 | × | 〇 | 〇 | 〇※2 |
| 3 | 会員登録 | 〇 | × | × | 〇 |
| 4 | 会員変更 | × | 〇 | 〇 | 〇※4 |
| 5 | 会員削除 | × | 〇 | 〇 | 〇※4 |

※1 ログインしている会員自身の情報のみを扱うことができる

※2 全ての会員の情報を閲覧することができる

※3 一般会員の情報を閲覧することができる

※4 一般会員の情報を変更、削除することができる

1. 技術的な制約・既知の問題点

(1) 入力チェックについて

必須入力の入力欄に、全角空白文字のみを入力しても、未入力と判断されない。

(2) 退会後の処理について

一般会員の退会後、完了画面でリロード(再読み込み)をするとエラーが発生する場合がある。

(3) ファイルアップロード機能について

画像のアップロード処理を実行した際に、画面上にアップロードしたファイルが即時に反映されない場合がある。この問題は、サーバーにアップロードされたファイルがEclipse中のデータ保存場所と同期されていないことが原因のため、対処を不要とする。なお、Eclipseからプロジェクトをリフレッシュした場合、問題なく反映される。