



BUSINESS CONSULTING

SOMMAIRE

Business consulting	1
Cahier des charges	5
Modele conceptuel de données.....	6
Modèle relationnel de données :.....	6
Schémas visuel	7
Architecture de l'application.....	8
Principe de développement et outils	9
Découpage modulaire	12
Module 1 : L'accueil.....	12
Module 2 : Compte.....	12
Module 3 : Projet.....	12
Module 4 : Phase.....	12
Module d'accueil.....	13
Représentation graphique :.....	13
Point 1	13
Point 2	13
Point 3	13
Point 4	14
Fichiers xhtml utilisés pour ce module en fonction des profils :.....	14
Module login	15
Représentation graphique :.....	15
Point 1	15
Point 2	15
Point 3	15
Fichiers xhtml utilisés pour ce module en fonction des profils :.....	15
Module compte.....	16
Représentation graphique :.....	16
Point 1	16
Point 2	16
Point 3	17

Point 4	17
Fichiers xhtml utilisés pour ce module en fonction des profils :	17
Module projet	18
Représentation graphique :	18
Point 1	18
Point 2	18
Point 3	19
Point 4	19
Point 5	19
Point 6	19
Point 7	20
Fichiers xhtml utilisés pour ce module en fonction des profils :	20
Module modification d'un projet	21
Représentation graphique :	21
Point 1	21
Point 2	21
Point 3	21
Point 4	22
Fichiers xhtml utilisés pour ce module en fonction des profils :	22
Module phase	23
Représentation graphique :	23
Point 1	23
Point 2	23
Point 3	24
Point 4	24
Point 5	24
Fichiers xhtml utilisés pour ce module en fonction des profils :	24
Module modification d'une phase	25
Représentation graphique :	25
Point 1	25
Point 2	25
Point 3	25
Point 4	26

Fichiers xhtml utilisés pour ce module en fonction des profils :	26
Module ajout d'une phase	27
Représentation graphique :	27
Point 1	27
Point 2	27
Point 3	27
Point 4	28
Fichiers xhtml utilisés pour ce module en fonction des profils :	28
Module ajout d'un ticket.....	29
Représentation graphique :	29
Point 1	29
Point 2	29
Point 3	29
Point 4	30
Fichiers xhtml utilisés pour ce module en fonction des profils :	30
Module modification d'un ticket.....	31
Représentation graphique :	31
Point 1	31
Point 2	31
Point 3	31
Point 4	32
Fichiers xhtml utilisés pour ce module en fonction des profils :	32

CAHIER DES CHARGES

Ce logiciel a pour vocation à améliorer la gestion de projets des chefs de projet. A terme elle devra permettre également l'édition de factures par un utilisateur ayant le rôle de comptable et la création de nouveaux clients, nouveaux projets par un utilisateur ayant le rôle d'administrateur.

Pour cette première version, chaque projet est créé directement dans la base de données avant d'être accessible par un chef de projet.

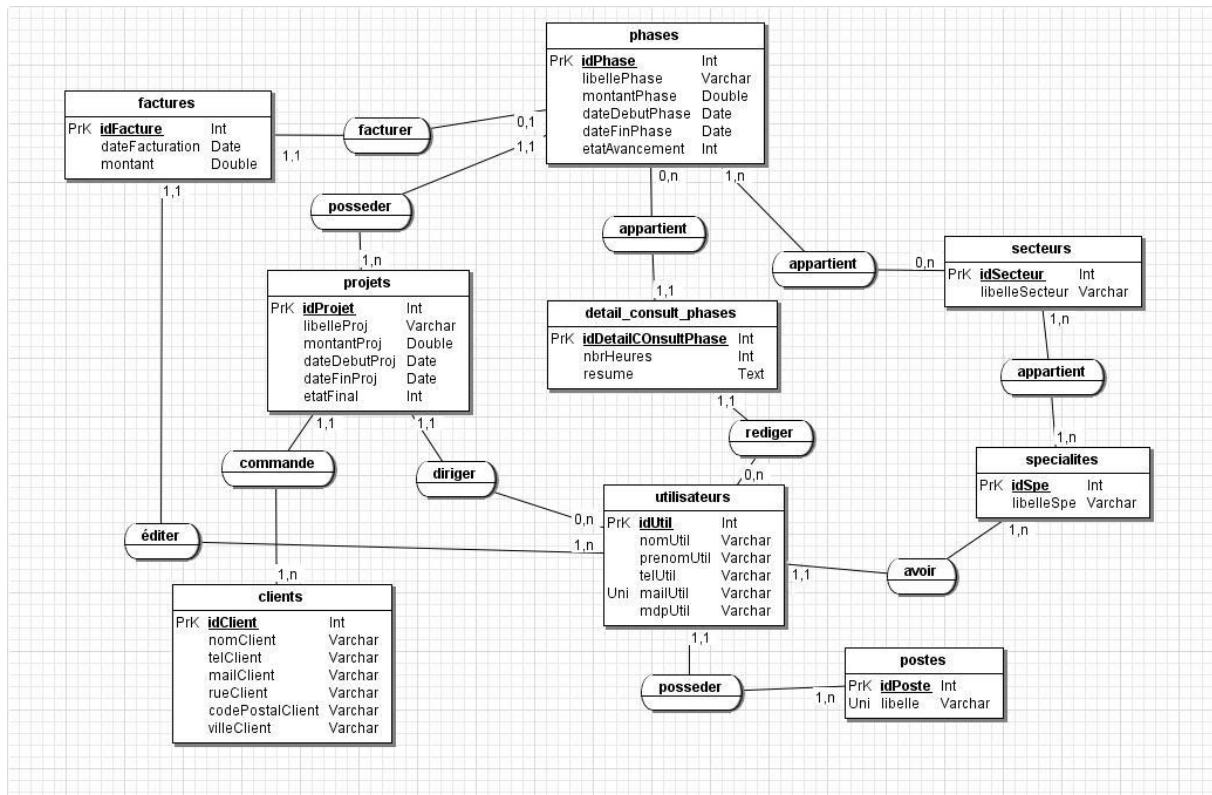
Un chef de projet aura la possibilité de :

- Visualiser les projets qu'il gère (état, temps passé etc.)
- Modifier un projet
- Créer des phases pour un projet
- Modifier des phases

Un consultant aura la possibilité de :

- Accéder aux projets qui auront le statut en cours
- Ajouter un ticket d'intervention si la phase correspond à son secteur d'activité et si cette dernière est ouverte.
- Modifier un ticket qu'il a lui-même créé

MODELE CONCEPTUEL DE DONNEES



Modèle relationnel de données :

projet (idProjet, libelleProj, montantProj, dateDebutProj, dateFinProj, etatFinal, #fk_idChefProj, #fk_idClt)

phase (idPhase, libellePhase, montantPhase, dateDebutPhase, dateFinPhase, etatAvancement, #fk_idProjet)

phase_sect (#idPhase, #idSecteur)

secteurs (idSecteur, libelleSecteur)

secteur_spe (#fk_idSecteur, #fk_idSpe)

specialites (idSpe, libelleSpe)

utilSpe (#idUtil, #idSpe)

utilisateurs (idUtil, nomUtil, prenomUtil, telUtil, mailUtil, mdpUtil)

util_postes (#fk_mailUtil, #fk_idPoste)

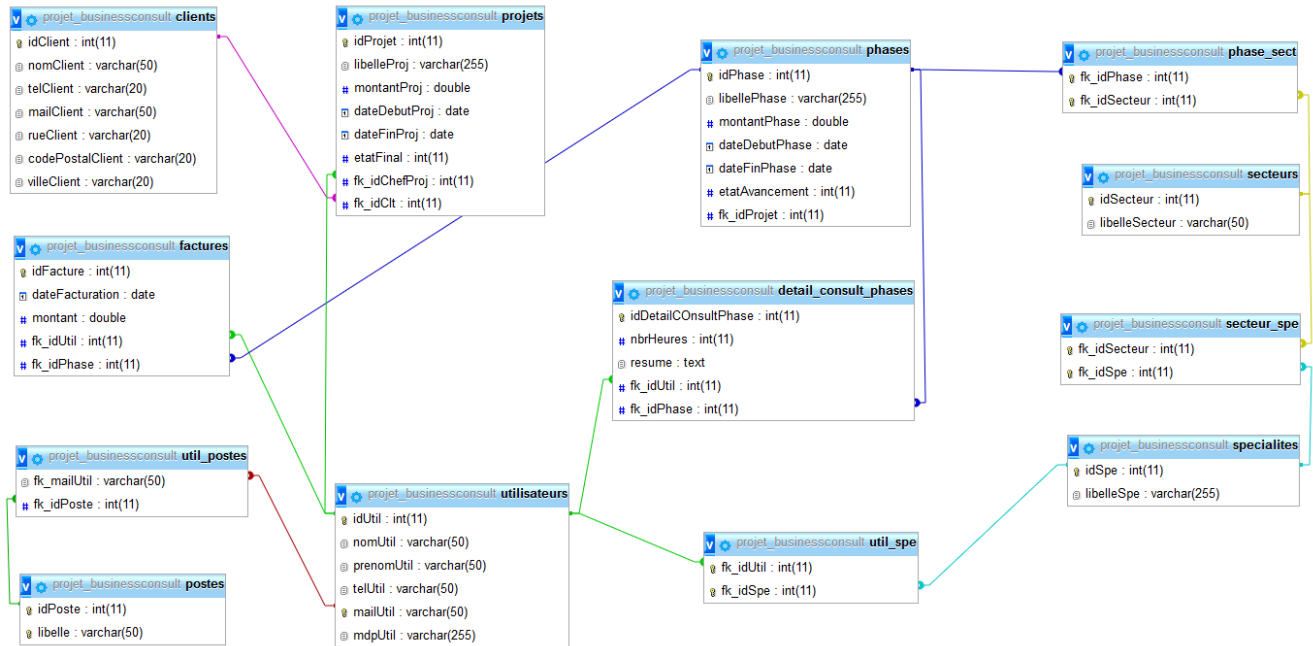
postes (idPoste, libelle)

factures (idFacture, dateFacturation, montant, #fk_idUtil, #fk_idPhase)

detail_consult_phases (idDetailConsultPhase, nbrHeures, resume, #fk_idUtil, #fk_idPhase)

clients (idClt, nomClient, telClient, mailClient, rueClient, codePostalClient, villeClient)

Schémas visuel

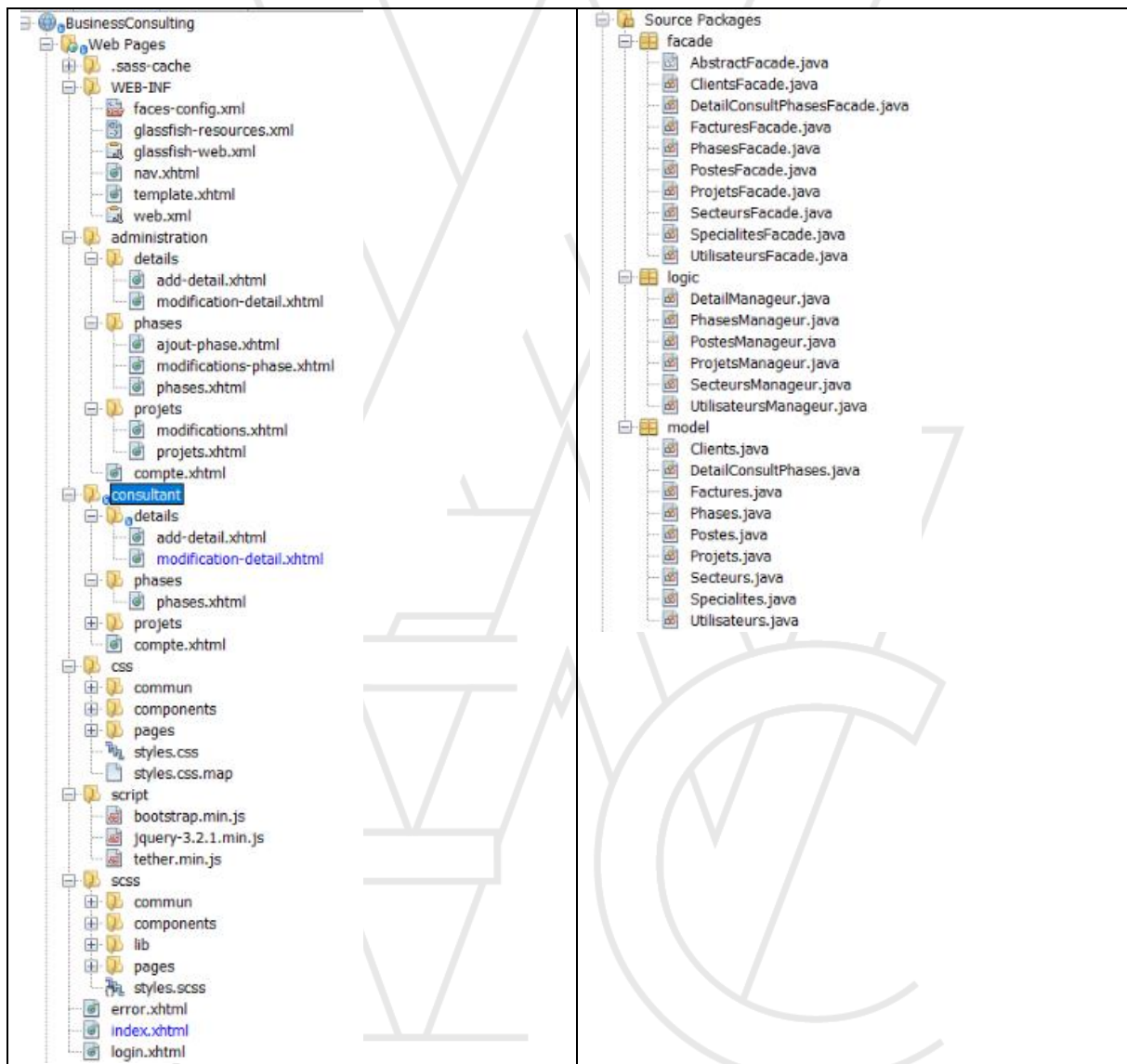


ARCHITECTURE DE L'APPLICATION

L'application a été développée sous java ee 7. Le pattern façade a été utilisé pour le développement de la partie logique de cette application et pour les pages xhtml c'est le Framework JSF 2.2 qui a été utilisé.

Partie front :

partie logique :



PRINCIPE DE DEVELOPPEMENT ET OUTILS

L'application a été développée sous le pattern Façade.

« Une bonne pratique de conception est d'essayer de limiter le couplage existant entre des fonctionnalités proposées par différentes entités. Dans la pratique, il est préférable de développer un petit nombre de classes et de proposer une classe pour les utiliser. C'est ce que propose le motif de conception façade.

Le but est de proposer une interface facilitant la mise en œuvre d'un ensemble de classes généralement regroupées dans un ou plusieurs sous-systèmes. Le motif Façade permet d'offrir un niveau d'abstraction entre l'ensemble de classes et celles qui souhaitent les utiliser en proposant une interface de plus haut niveau pour utiliser les classes du sous-système.

Le modèle Façade peut être utilisé pour :

- Faciliter l'utilisation partielle d'un sous-système complexe ou de plusieurs classes
- Masquer l'existence d'un sous-système
- Ajouter des fonctionnalités sans modifier le sous-système
- Assurer un découplage entre le client et le sous-système (par exemple pour chaque couche d'une architecture logicielle N tiers)

L'utilisation d'une façade permet au client de limiter le nombre d'objets à utiliser puisqu'il se contente simplement d'appeler une ou plusieurs méthodes de la façade. Ce sont ces méthodes qui vont utiliser les classes du sous-système, masquant ainsi au client toute la complexité de leur mise en œuvre.

Il peut être pratique de définir une façade sans état (les méthodes de la façade n'utilisent pas de membres statiques de la classe) car dans ce cas, une seule et unique instance de la façade peut être définie côté client en mettant en œuvre le modèle de conception singleton prévu à cet effet.

Il est possible de proposer des fonctionnalités supplémentaires dans la façade qui enrichissent la mise en œuvre du sous-système.

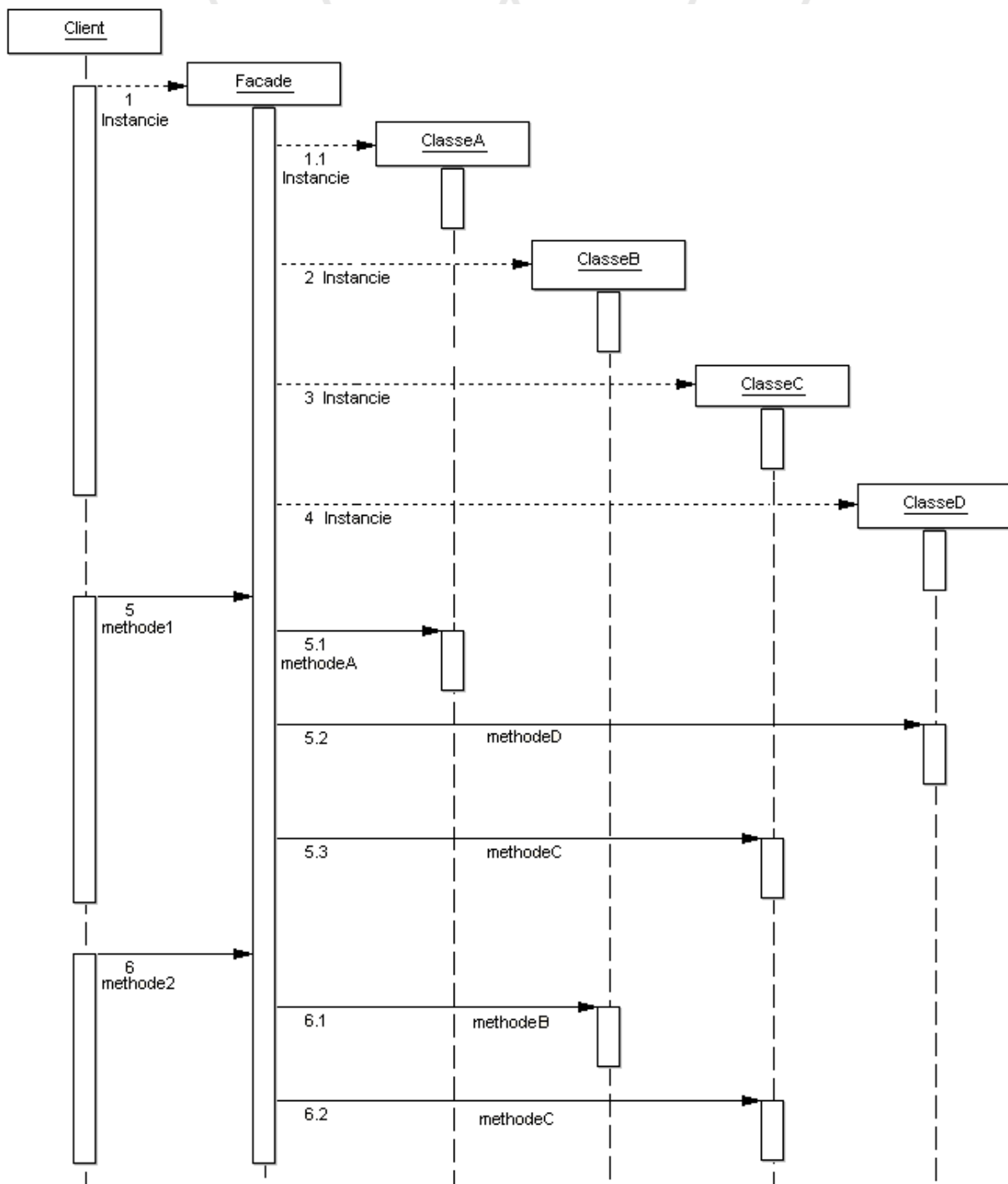
La façade peut aussi être utilisée pour masquer le sous-système. Elle peut encapsuler les classes du sous-système et ainsi cacher au client l'existence du sous-système. Cette mise en œuvre facilite le remplacement du sous-système par un autre : il suffit simplement de modifier la façade pour que le client continue à fonctionner.

Il est possible que toutes les fonctionnalités proposées par les classes du sous-système ne soient pas accessibles par la façade : son but est de simplifier leurs utilisations mais pas de proposer toutes les fonctionnalités.

Ce motif de conception est largement utilisé. »

Explication de ce pattern par Jean Michel Doux sur

<https://www.jmdoudoux.fr/java/dej/chap-design-patterns.htm>



Concernant la partie xhtml j'ai utilisé le framework JSF 2.2 car c'est l'un des framework les plus utilisés en ce moment. Il offre de nombreuses possibilités sur les pages xhtml grâce à l'utilisation de balises JSF.

L'ensemble des pages de cette application respecte les normes des w3c.

Concernant le serveur, l'application a été développée sur un serveur local Glassfish et développée sur avec l'IDE netbeans 8.2.

La base de données utilisée est MySQL.



DECOUPAGE MODULAIRE

Module 1 : L'accueil

Au chargement de l'application, l'utilisateur est dirigé vers la page d'accueil (**module accueil**). Sur celui-ci, il peut visualiser un message d'accueil et 3 informations. Il a le choix entre se connecter via le menu ou aller sur la page login (**module login**).

Module 2 : Compte

L'utilisateur est connecté. En fonction de son statut il est redirigé vers le dossier administration ou consultant sur la page compte (**module compte**).

- S'il a le statut de chef projet, il visualise tous les projets qu'il dirige ou a dirigé.
- S'il a le statut de consultant, il ne voit que les projets en cours.

Module 3 : Projet

Le projet est chargé correctement (**module projet**). L'utilisateur voit toutes les phases du projet.

- S'il est chef de projet il peut :
 - Modifier le projet en cours (**module modification de projet**).
 - Ajouter une phase (**module ajout d'une phase**).
 - Voir une phase
- S'il est consultant il ne peut que choisir la phase qu'il souhaite voir.

Module 4 : Phase

La phase correctement chargée (**module phase**). L'utilisateur peut alors, si cette dernière n'est pas terminée :

- S'il est chef de projet :
 - Modifier la phase (**module modification d'une phase**).
 - Ajouter un ticket (**module ajout d'un ticket**).
 - Modifier un ticket qu'il a lui-même créé (**module modification d'un ticket**).
- S'il est consultant :
 - Ajouter un ticket
 - Modifier un ticket qu'il a lui-même créé

Module d'accueil

Représentation graphique :



Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Il s'agit du formulaire de login.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java
- facade/UtilisateursFacade.java

Point 3

Visible par tous les utilisateurs. Il s'agit du listing des phases en cours de réalisations.

Classes utilisées :

- model/Phases.java

- logic/ PhasesManageur.java
- facade/PhasesFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : nav.xhtml ; index.xhtml ; template.xhtml



Module login

Représentation graphique :

The screenshot shows a web page layout with three numbered points of interest:

- Point 1:** A dark grey navigation bar at the top containing the text "Label Label".
- Point 2:** A login form in the center containing the text "Lorem", an "Email Address" input field, a "password" input field, and a "submit" button.
- Point 3:** A dark grey footer bar at the bottom.

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous. Il s'agit du formulaire de connexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java
- facade/UtilisateursFacade.java

Point 3

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

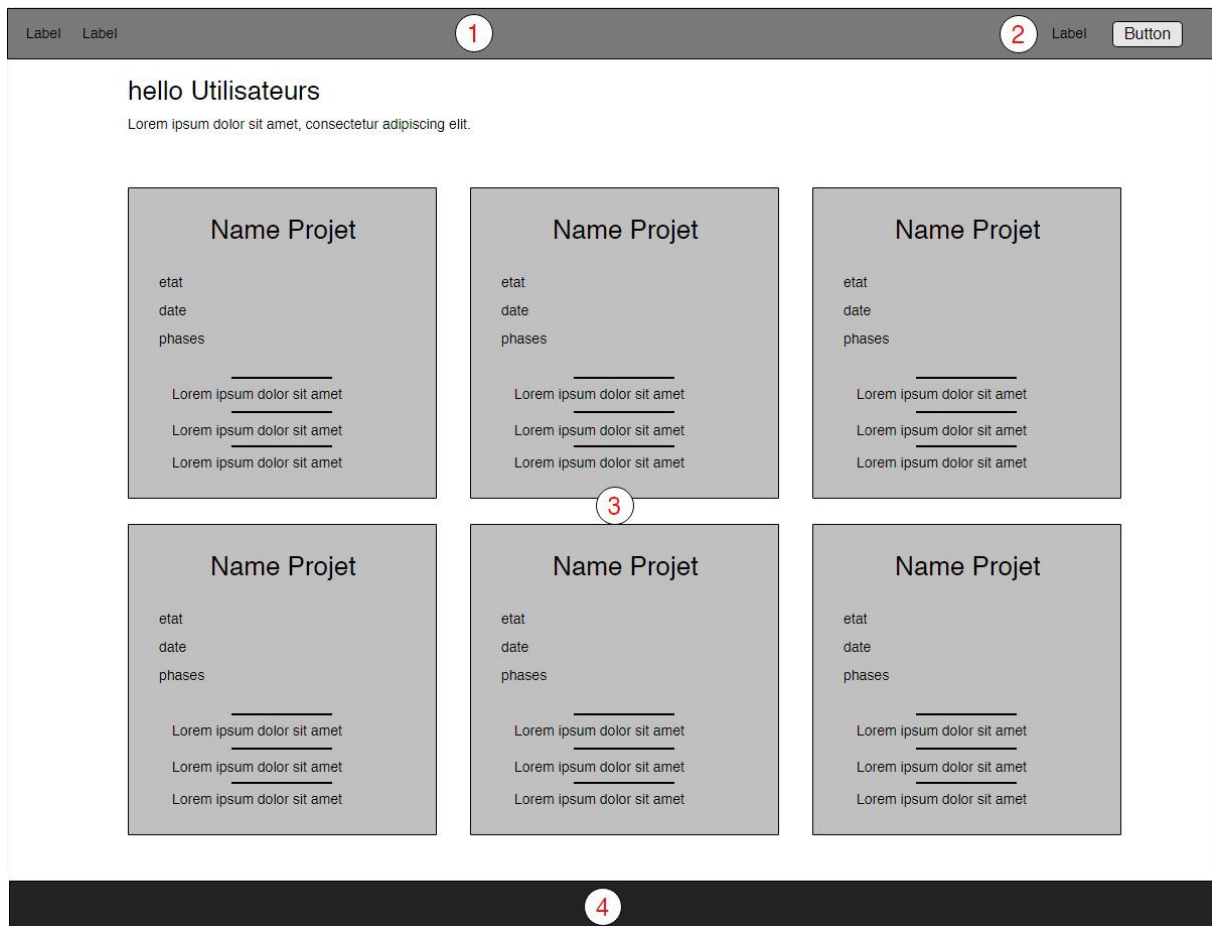
- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : nav.xhtml ; login.xhtml ; template.xhtml

Module compte

Représentation graphique :



Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnection.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Cette zone change en fonction du statut de l'utilisateur.

- S'il a le statut de chef projet, il visualise tous les projets qu'il dirige ou a dirigé.
- S'il a le statut de consultant, il ne voit que les projets en cours.

Classes utilisées :

- model/Utilisateurs.java
- model/Projets.java
- model/Phases.java
- logic/UtilisateursManageur.java
- logic/ProjetsManageur.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

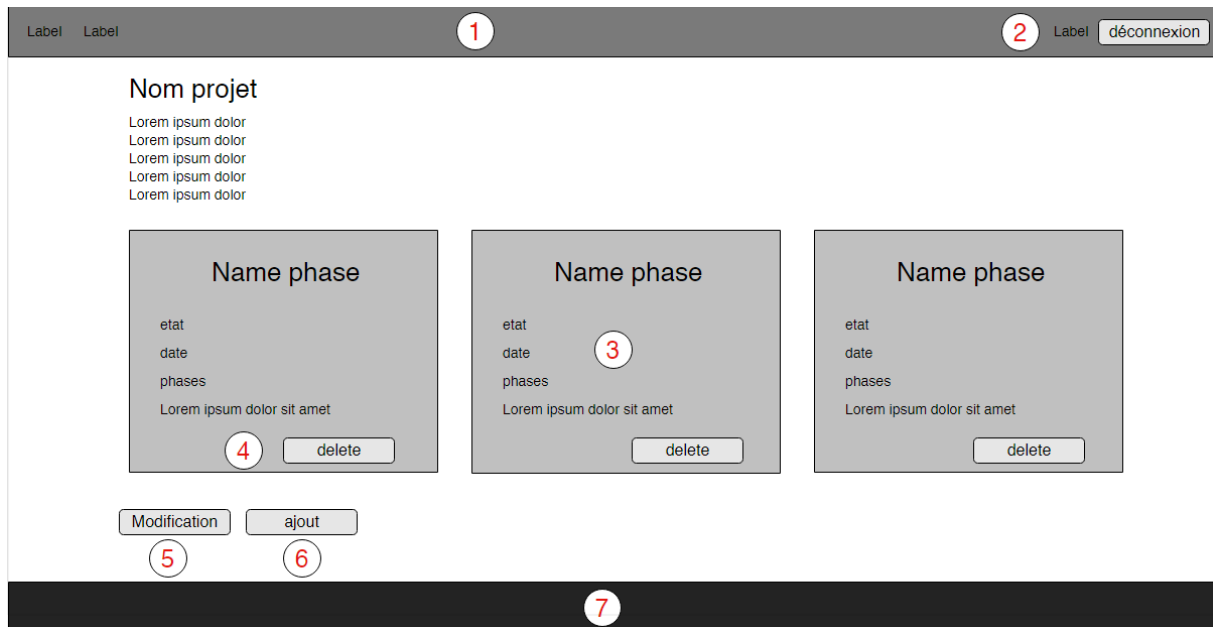
Pour tous les profils : template.xhtml ; nav.xhtml ;

Pour les chefs de projets : administration/compte.xhtml ;

Pour les consultants : consultant/compte.xhtml ;

Module projet

Représentation graphique :



Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManager.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManager.java

Point 3

Visible par tous cette zone liste les phases du projet et permet d'accéder au ticket de cette dernière en cliquant dessus.

Classes utilisées :

- model/Projets.java
- model/Phases.java
- model/Utilisateurs.java
- logic/ProjetsManageur.java
- logic/UtilisateursManageur.java
- facade/ProjetsFacade.java

Point 4

Visible uniquement par les utilisateurs ayant le statut de chef de projet. Cette zone au sein du point 3 permet de supprimer une phase d'un projet.

Classes utilisées :

- model/Phases.java
- logic/PhasesManageur.java
- facade/PhasesFacade.java

Point 5

Visible uniquement par les utilisateurs ayant le statut de chef de projet. Cette zone permet de naviguer vers la page de modification de projet.

Classes utilisées :

- Aucune

Point 6

Visible uniquement par les utilisateurs ayant le statut de chef de projet. Cette zone permet de naviguer vers la page d'ajout de phase pour le projet en cours de visionnage.

Classes utilisées :

- Aucune

Point 7

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/projets/projets.xhtml

Pour les consultants : consultant/projets/projets.xhtml



Module modification d'un projet

Représentation graphique :

The screenshot shows a web application interface for project modification. It features a dark navigation bar at the top with the text 'Label Label' and a 'déconnexion' button. Below the navigation bar, the main content area is titled 'Nom du projet (modification)'. This area contains four 'Label' text boxes, each followed by an 'input' field. A 'retour' button is located to the left of the input fields, and a 'submit' button is located to the right. The interface is annotated with four numbered points: 1. The navigation bar, 2. The navigation bar, 3. The form area, and 4. The dark footer bar.

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManager.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManager.java

Point 3

Visible uniquement par les chefs de projets. Formulaire de modification d'un projet.

Classes utilisées :

- model/Utilisateurs.java
- model/Projets.java
- logic/ProjetsManager.java
- logic/UtilisateursManager.java
- facade/ProjetsFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

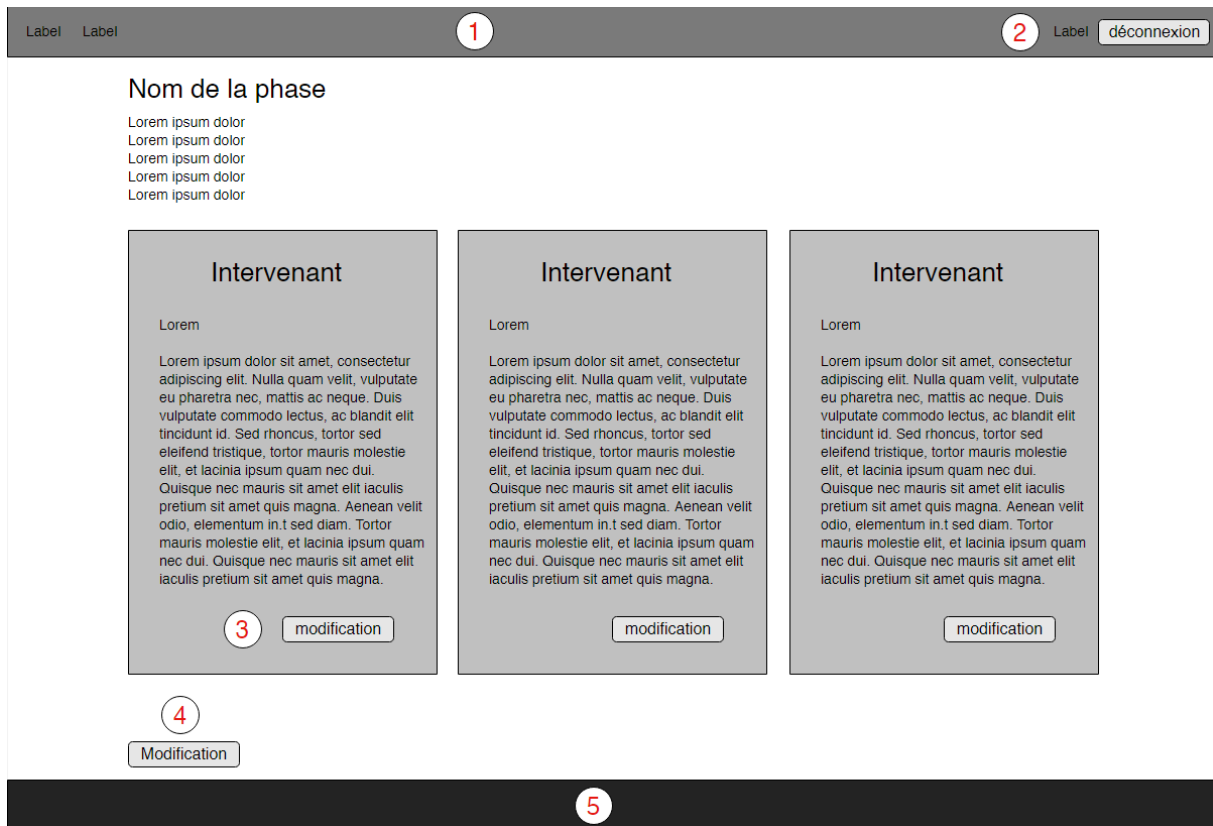
Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/projets/modifications.xhtml



Module phase

Représentation graphique :



Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Visible par tous les utilisateurs. Cette zone liste tous les tickets d'interventions sur la phase.

- Cette zone change en fonction du statut de l'utilisateur. model/Phases.java
- model/Projets.java
- model/Utilisateurs.java
- logic/PhasesManageur.java
- logic/ProjetsManageur.java
- logic/UtilisateursManageur.java
- facade/PhasesFacade.java

Point 4

Visible uniquement par les utilisateurs ayant le statut de chef de projet. Cette zone permet de naviguer vers la page de modification de la phase.

Classes utilisées :

- Aucune

Point 5

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

Aucune.

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/phases/phases.xhtml

Pour les consultants : consultant/phases/phases.xhtml

Module modification d'une phase

Représentation graphique :

Label Label 1 2 Label déconnexion

Nom de la phase (modification)

Label input

Label input

3 Label input

Label input

retour submit

4

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Cette zone change en fonction du statut de l'utilisateur.

- model/Phases.java
- model/Secteurs.java
- model/Utilisateurs.java
- logic/SecteurManageur.java
- logic/PhasesManageur.java
- logic/UtilisateursManageur.java
- facade/SecteursFacade.java

- facade/PhasesFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour les chefs de projets : template.xhtml; nav.xhtml ; administration/phases/modifications-phase.xhtml

Module ajout d'une phase

Représentation graphique :

The screenshot shows a web application interface for adding a phase to a project. The interface is divided into four numbered points:

- Point 1:** Navigation bar with 'Label Label' and a 'déconnexion' button.
- Point 2:** Header area with the title 'Ajout d'une phase au "nom de projet"'. Below the title, there are four 'Label' inputs and a 'submit' button.
- Point 3:** Form area with four 'Label' inputs and a 'retour' button.
- Point 4:** Footer bar.

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Visible uniquement par les chefs de projets. Formulaire d'ajout d'une phase.

Classes utilisées :

- model/Phases.java
- model/Secteurs.java
- model/Utilisateurs.java
- model/Projets.java
- logic/SecteurManageur.java
- logic/PhasesManageur.java
- logic/UtilisateursManageur.java

- logic/ProjetsManageur.java
- facade/SecteursFacade.java
- facade/PhasesFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

- Aucune

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/phases/ajout-phase.xhtml

Module ajout d'un ticket

Représentation graphique :

The screenshot shows a web application interface. At the top, there is a dark header bar containing two 'Label' placeholders and a 'déconnexion' button, marked with a red circle '2'. Below the header, the main content area has a title 'Ajout d'un ticket à la phase "nom de la phase"'. Underneath the title, there is a 'Label' followed by an 'input' field. To the left of the input field is a red circle '3'. Below the input field is a 'Multi-line textarea'. At the bottom of the main content area, there are two buttons: 'retour' on the left and 'submit' on the right. At the very bottom, there is a dark footer bar marked with a red circle '4'.

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Visible par tous les utilisateurs. Permet d'ajouter un ticket à la phase en cours.

Classes utilisées :

- model/Utilisateurs.java
- model/DetailConsultPhases.java
- model/Phases.java
- logic/UtilisateursManageur.java
- logic/DetailManageur.java
- logic/PhasesManageur.java ;
- facade/DetailConsultPhasesFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

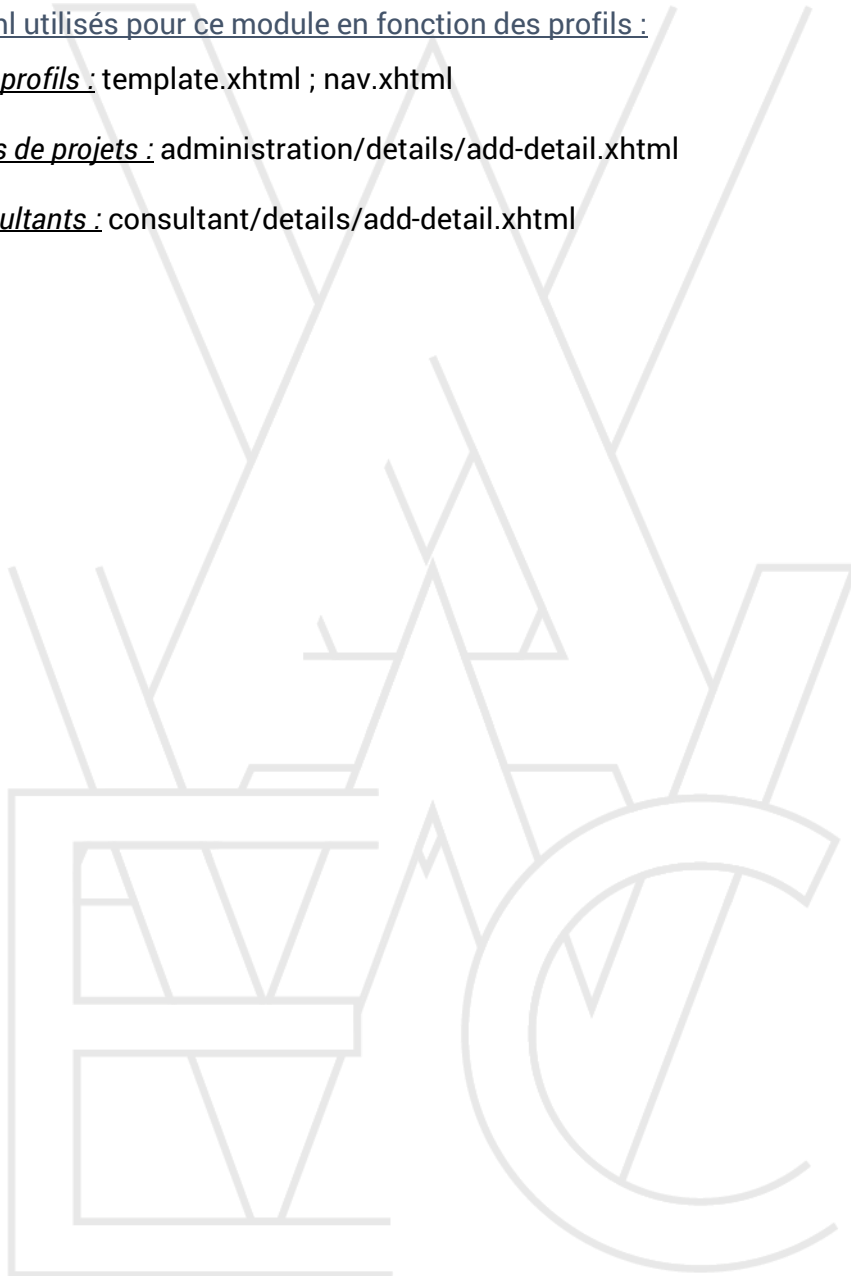
Aucune.

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/details/add-detail.xhtml

Pour les consultants : consultant/details/add-detail.xhtml



Module modification d'un ticket

Représentation graphique :

The screenshot shows a web application interface for modifying a ticket. It features a top navigation bar (Point 1) with the text 'Label Label' and a 'déconnexion' button (Point 2). The main content area (Point 3) is titled 'Modification ticket n°' and 'nom de la phase'. It contains an 'input' field, a 'Multi-line textarea', and two buttons: 'retour' and 'submit'. A bottom navigation bar (Point 4) is also visible.

Point 1

Visible par tous les utilisateurs. Il s'agit de la navigation.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 2

Visible par tous les utilisateurs. Cette zone affiche le nom de l'utilisateur et le bouton de déconnexion.

Classes utilisées :

- model/Utilisateurs.java
- logic/UtilisateursManageur.java

Point 3

Visible uniquement par le créateur du ticket. Permet de modifier le ticket créé.

Classes utilisées :

- model/Utilisateurs.java
- model/DetailConsultPhases.java
- model/Phases.java
- logic/UtilisateursManageur.java
- logic/ProjetsManageur.java
- logic/DetailManageur.java

- logic/PhasesManageur.java
- facade/DetailConsultPhasesFacade.java

Point 4

Visible par tous les utilisateurs. Il s'agit du pied de page.

Classes utilisées :

Aucune.

Fichiers xhtml utilisés pour ce module en fonction des profils :

Pour tous les profils : template.xhtml ; nav.xhtml

Pour les chefs de projets : administration/details/modification-detail.xhtml

Pour les consultants : consultant/details/modification-detail.xhtml