

De la spécification à la conception en UML

Juste un survol

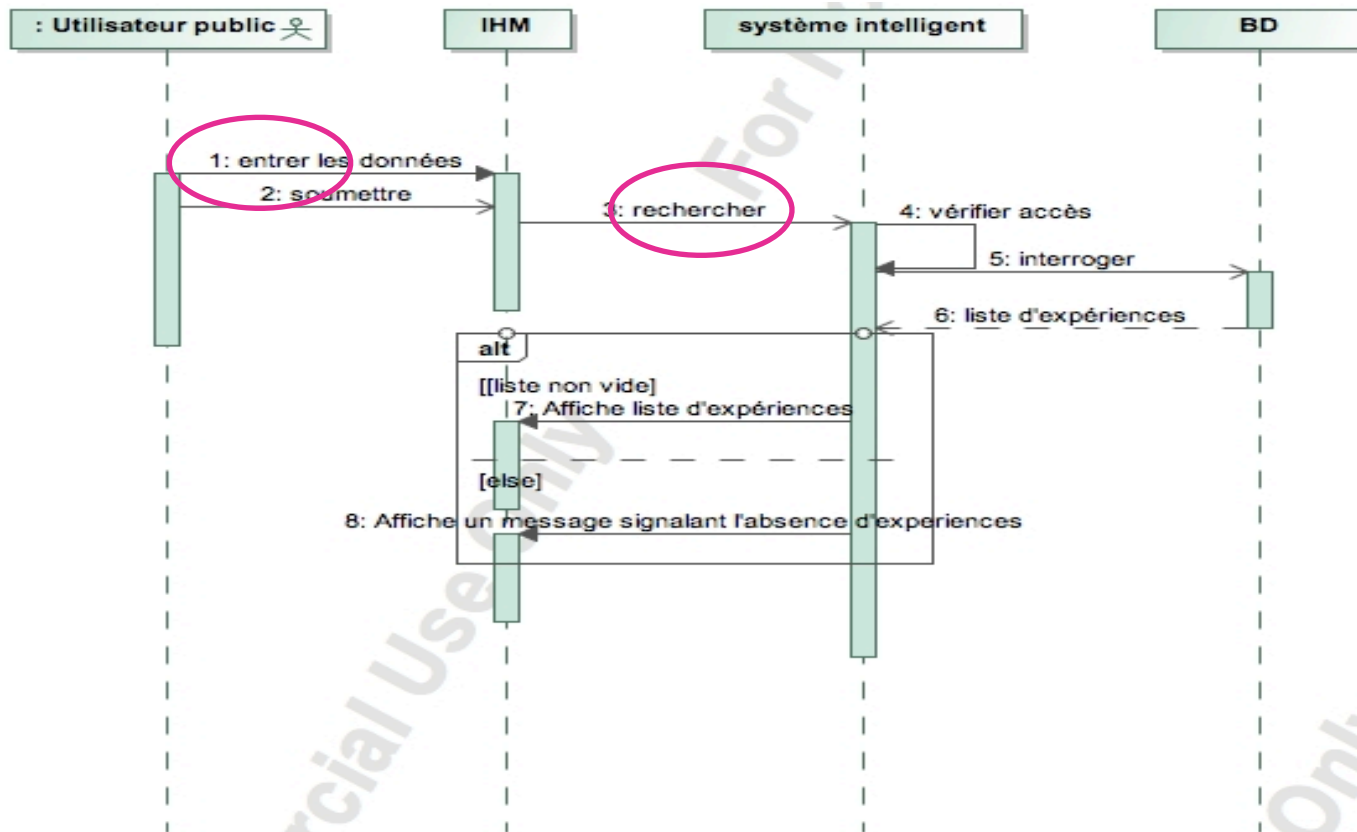
(voir UML en action pour en savoir plus par exemple)

Base d'implémentation ciblée :
Php et base de données

Etude des scenarii (1ère partie)

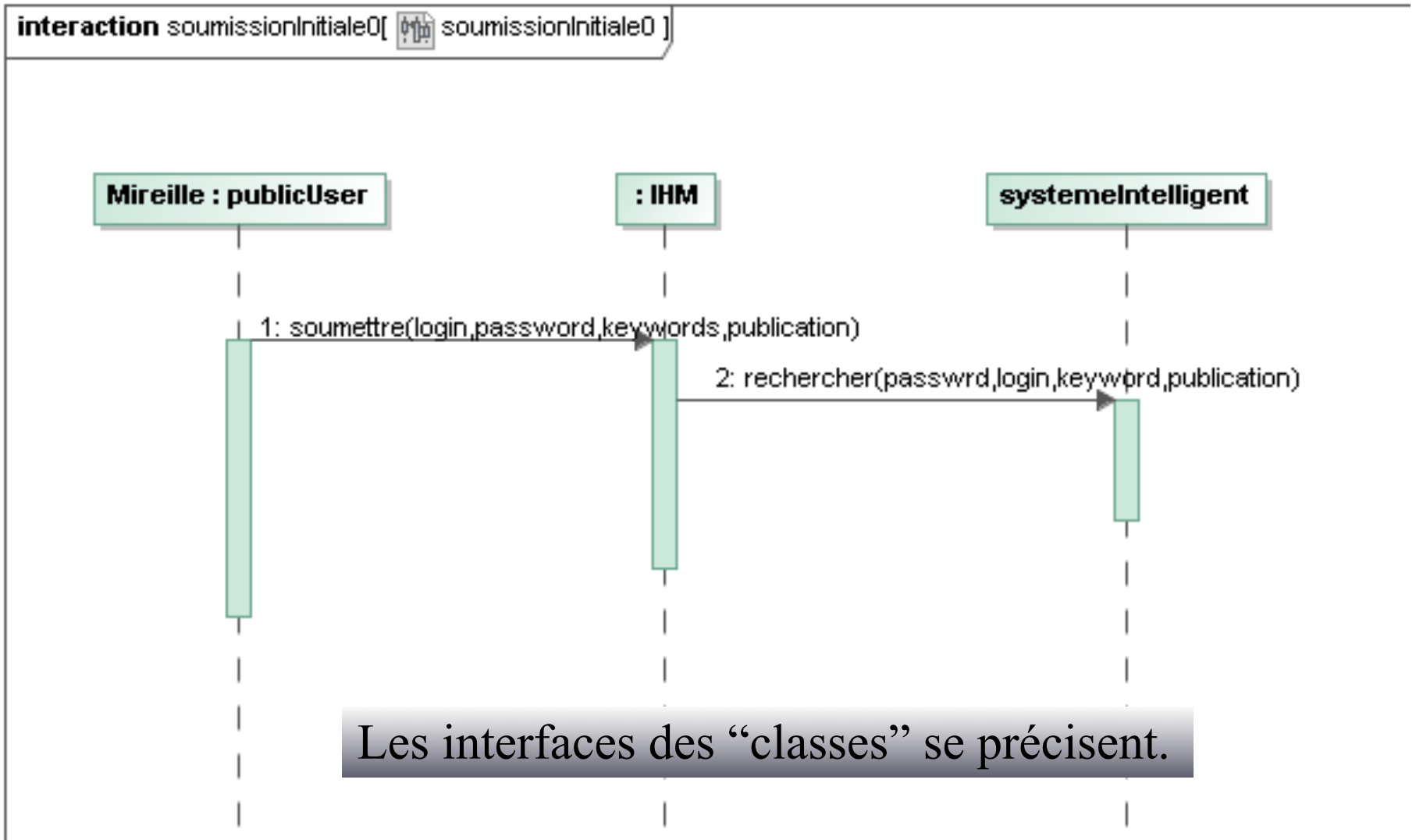
- Notion d'”objets applicatifs”
- Sessions ?
- Identification des bases de données
- Identification des interfaces utilisateur

Scénario : raffinement

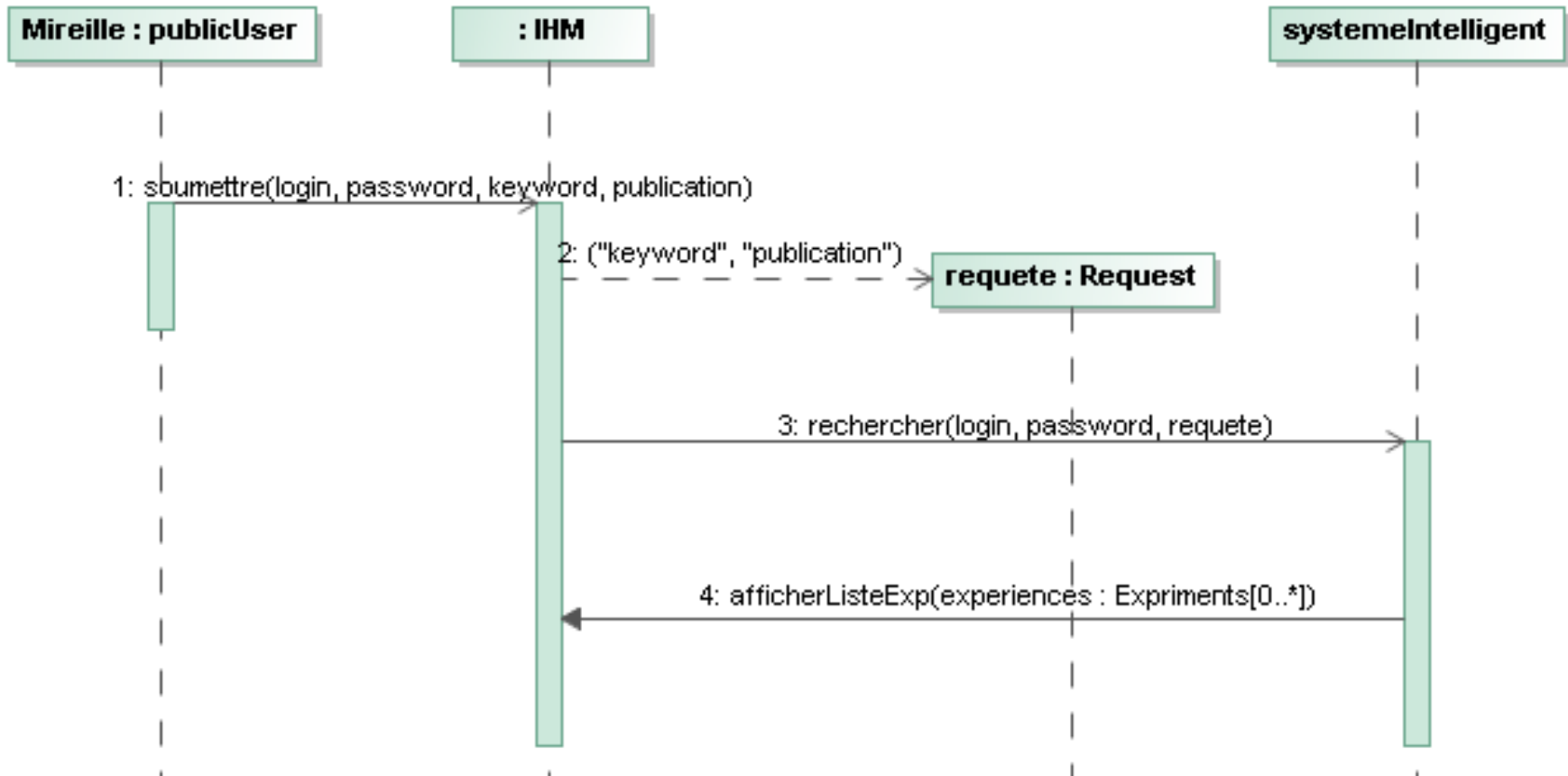


Quelles sont les informations pertinentes ?

Scénario : précisions

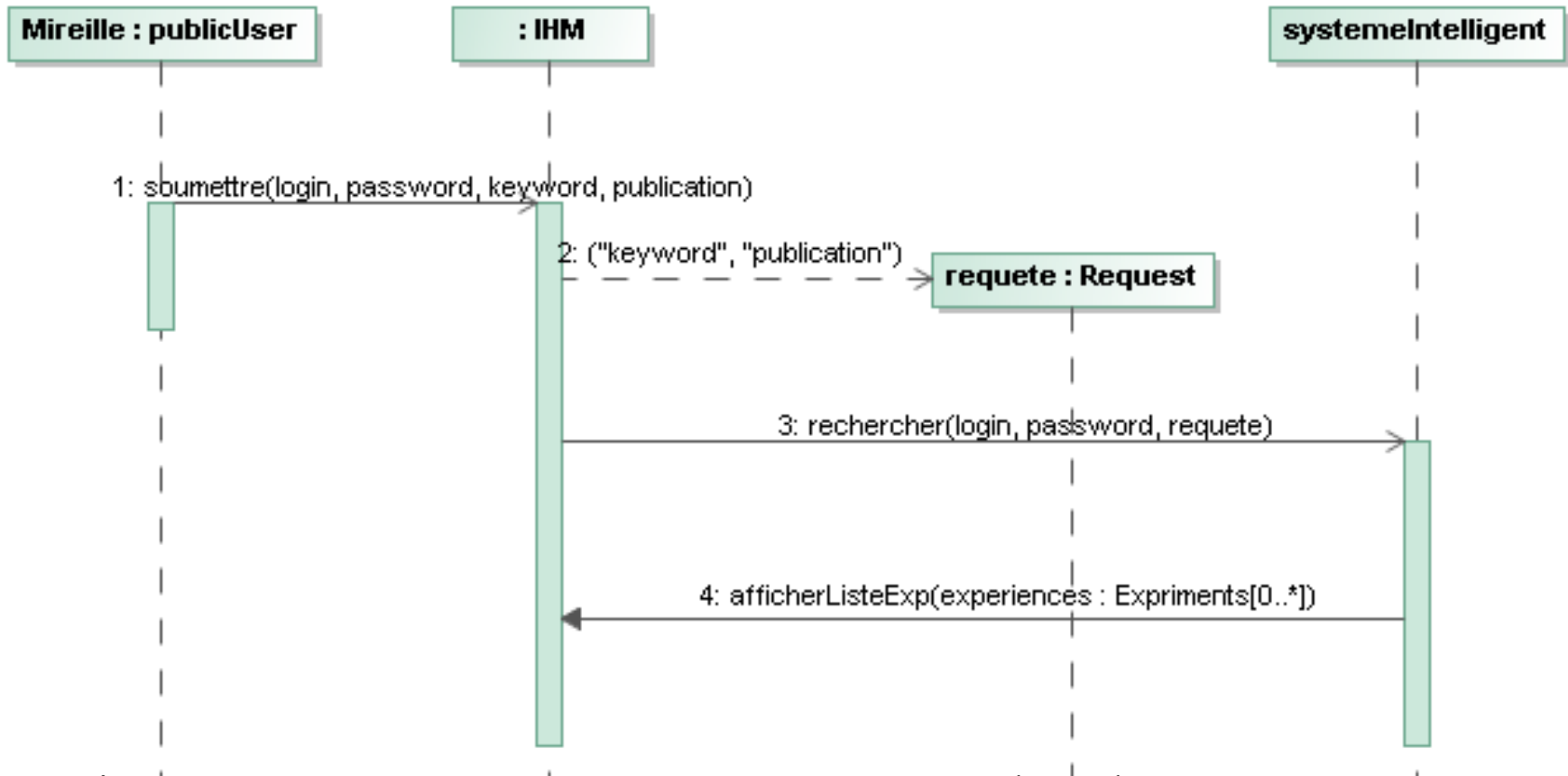


Scénario et réification



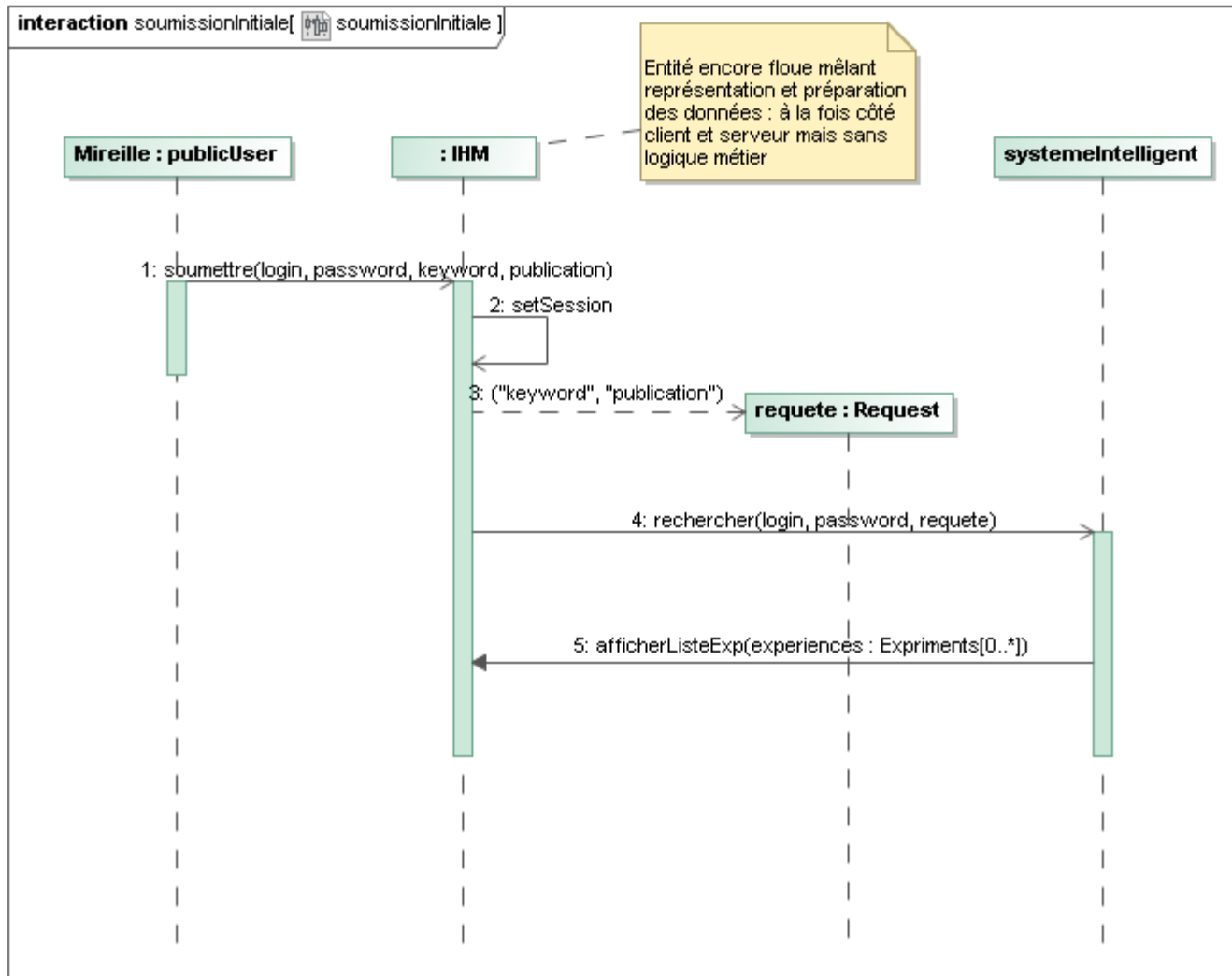
Des “classes” d’implémentation sont identifiées.

Scenarior et réification



Si à présent on veut ressoumettre une recherche...
Il faut retaper le mot de passe ?

Scénario et introduction de session



Sessions

- But: conserver des valeurs d'une page à l'autre (utile dans les formulaires de commande en ligne)
- Fonctions et variables PHP:
 - `session_start();` // en début de page pour indiquer que l'on travaille avec une session
 - `session_register("mavar");` // indiquer initialement que la variable *mavar* sera accessible d'une page à l'autre
 - `session_is_registered("mavar");` // retourne True si *mavar* a été déclarée comme variable de session
 - `session_id();` // indique le numéro d'identification (unique) de la session
 - `$_SESSION["mavar"]` // accès à une variable de session

PHP – Sessions, exemple

Attention
1^{ère} ligne
du fichier php

Page1.php

```
<?  
session_start (); // crée la session si inexistante  
$valeur = 10;  
session_register ("valeur"); // stocke $valeur dans la session  
$valeur = 20;  
?>
```

Page2.php

```
<?  
session_start ();  
echo $valeur; //affiche 20.  
?>
```

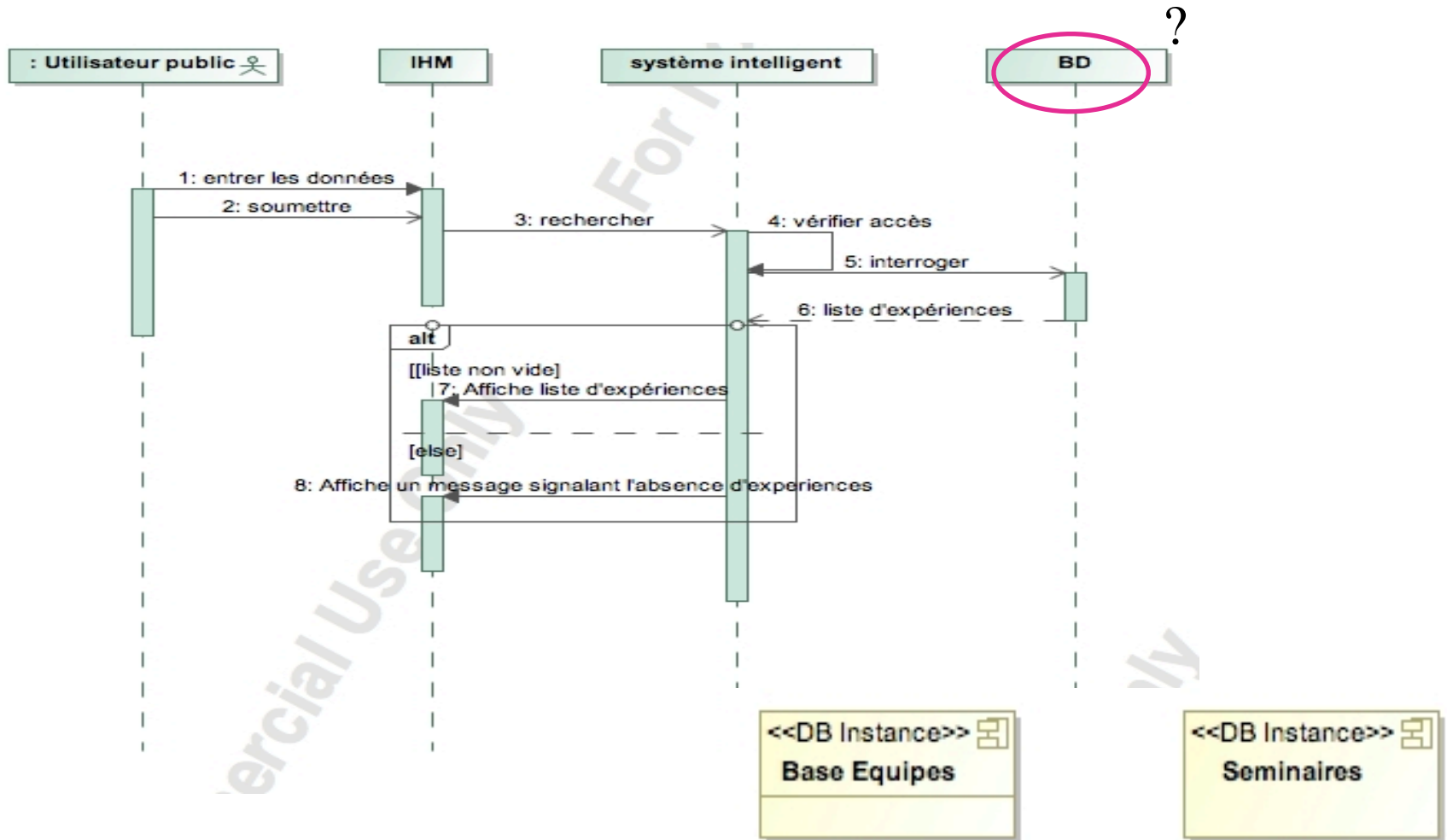
Sessions: exemple

```
<?php
    session_start(); // on indique que l'on travaille avec une session
    if (!session_is_registered("count")) {
        session_register("count");
        session_register("start");

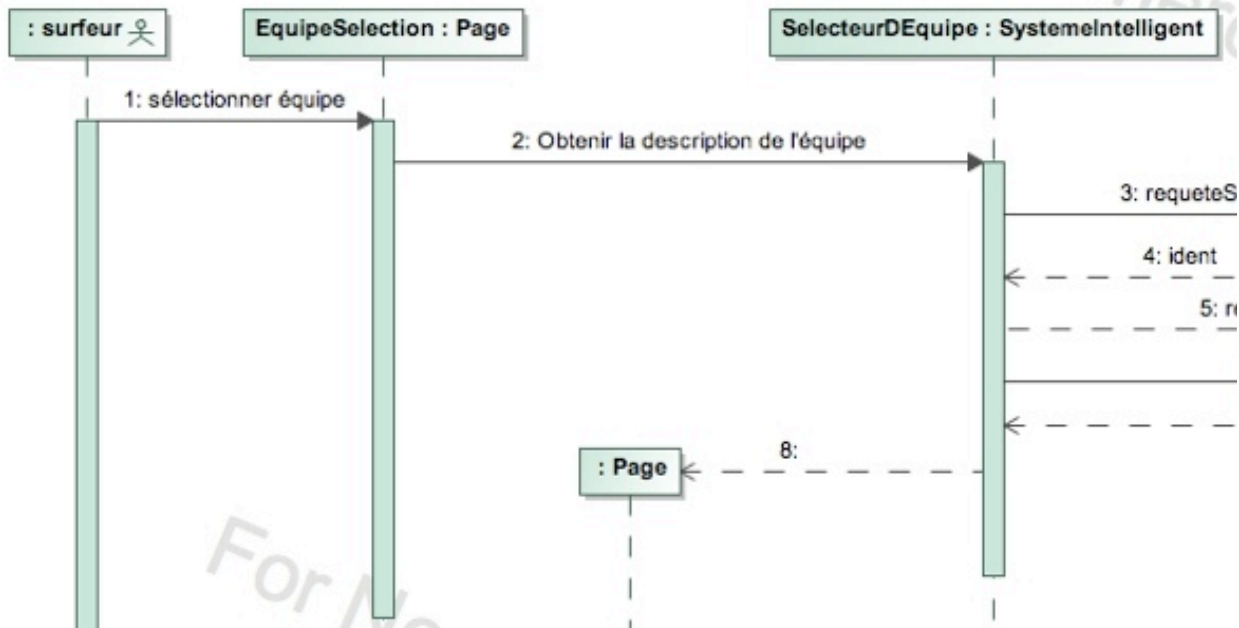
        $_SESSION["count"] = 0;
        $_SESSION["start"] = time();
    } else
        $_SESSION["count"]++;
    $sessionId = session_id();
?>

<html>
<head> <title>Session State Test</title></head>
<body>
    <p>This page points at a session (<?php echo $sessionId; ?>)
    <br>count = <?php echo $_SESSION["count"];?>.
    <br>start = <?php echo $_SESSION["start"];?>.
</body>
</html>
```

Scénario : identification des bases de données

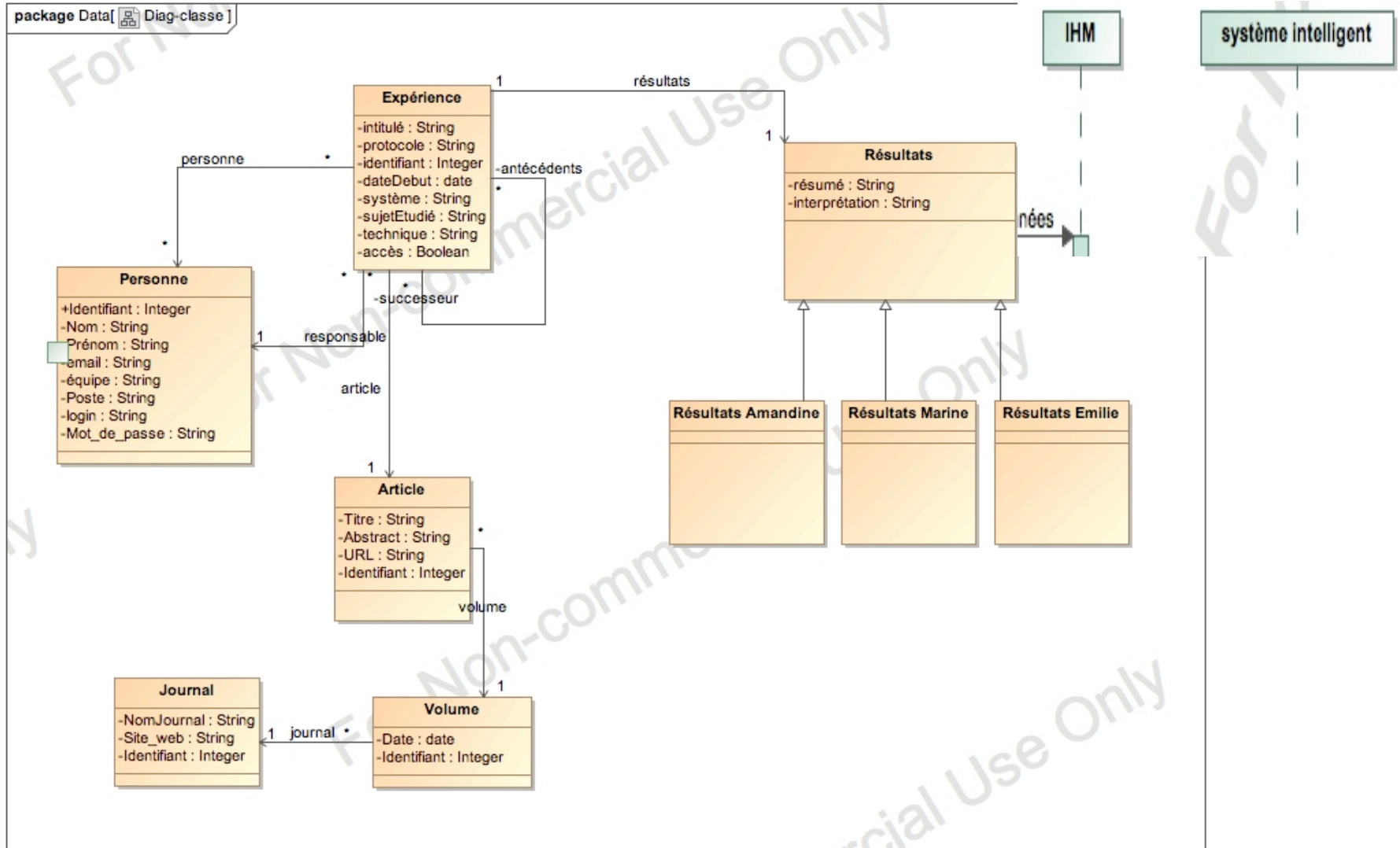


Scenario & use cases : Identification des interfaces utilisateur



Vue IHM	Description
Sélection Equipe	Sélection d'une équipe dans une liste triée des équipes du laboratoire ou par des critères de recherche.....
Visualisation d'une équipe	Visualisation d'une équipe par ses membres, ...

Etude du diagramme de classes

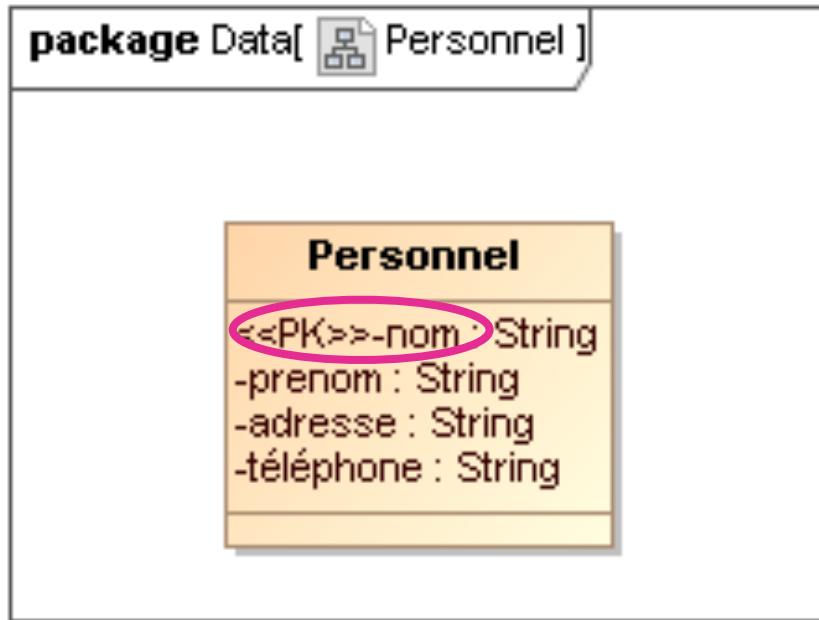


Conversion d'un modèle de classes en schéma relationnel

- À partir d'un diagramme de classes décrit en UML,
 - On sélectionne les classes qui correspondent à des données persistantes
 - Puis on passe au schéma relationnel correspondant.
- L'objectif est donc de :
 - représenter toutes les informations présentes dans le diagramme des classes sélectionné (classes et associations) ;
 - éviter les redondances (répétition des mêmes associations) ;
 - limiter le nombre de relations (schéma complexe) ;
 - limiter les valeurs absentes (e.g., les valeurs NULL).
- <http://www.math-info.univ-paris5.fr/~pellier/teaching/others/bd/cours/Modelisation.pdf>

Classe ...

- Implémentations des attributs, généralisations, et associations sous forme de tables



```
--
-- Structure de la table `personnel`
--
CREATE TABLE `personnel` (
    `nom` varchar(15) NOT NULL default "",
    `prenom` varchar(15) default NULL,
    `adresse` longtext,
    `telephone` varchar(12) default NULL,
    PRIMARY KEY (`nom`)
);
```

Identification des clefs primaires (ici utilisation de stéréotype)

Classes... Identification des clefs

Du modèle objet ... aux BD

Classes

Schéma relationnel

Attributs

Ajout de l'identifiant

ADRESSE
N°
Rue
CP
Ville

Schéma Adresse

Clé primaire

Id_adresse

Attribut	Domaine	Non Null
<u>Id_adresse</u>	Identifiant	Oui
N°	Entier	Non
Rue	String(30)	Non
.....		

Identifiant
génééré

Classes... Identification des clefs

Du modèle objet ... aux BD

Classes

Schéma relationnel

Attributs

Ajout de l'identifiant

PERSONNE
NSS {unique}
Nom
Prénom
Date-naissance

Schéma Personne

Clés primaires

Candidates

Id_adresse

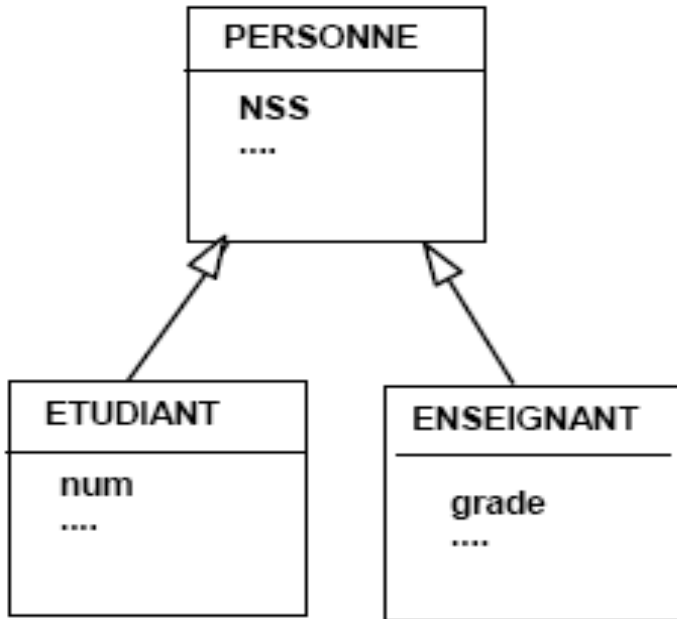
NSS

On préfère NSS

Attribut	Domaine	Non Null
<u>Id_Pers</u>	Identifiant	Oui
<u>NSS</u>	String(13)	Oui
Prénom	String(30)	Non
Date-nais	Date	Non

Héritage

Spécialisation

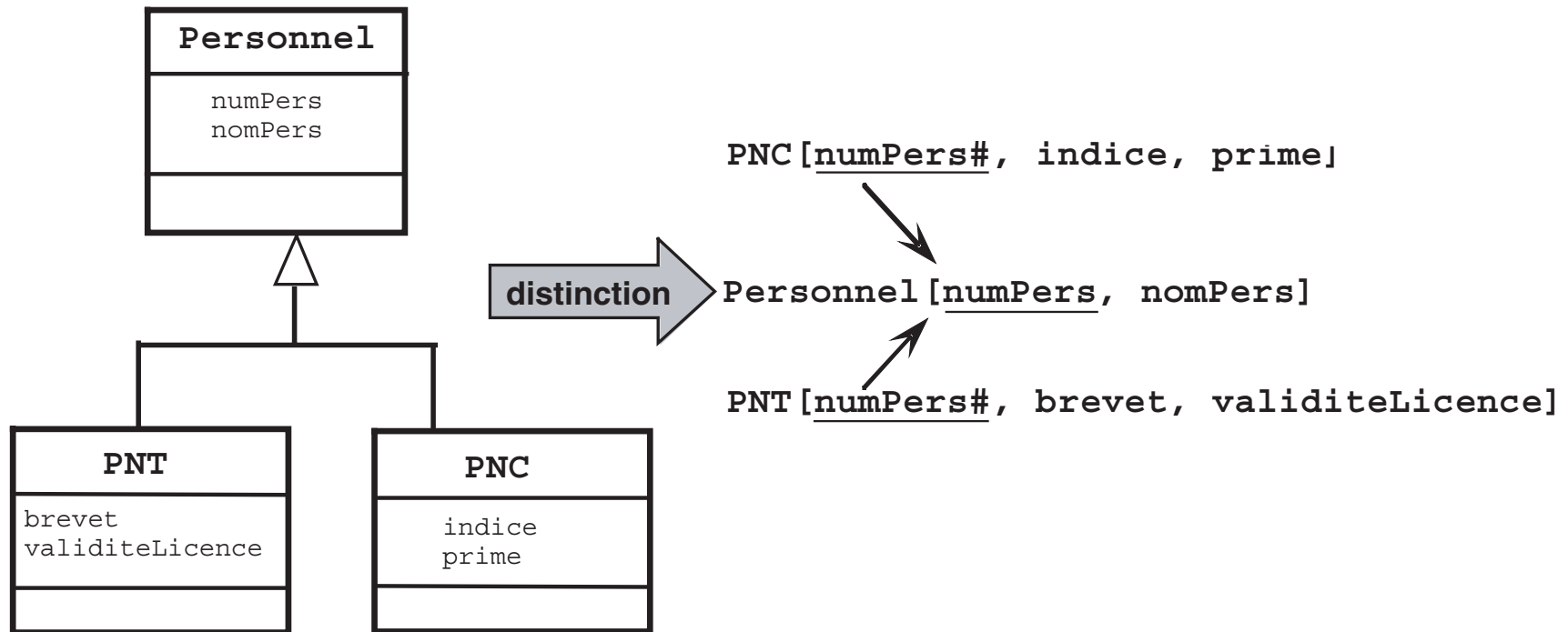


- Trois décompositions sont possibles
- décomposition par distinction
 - décomposition descendante
 - décomposition ascendante

UML 2 pour les bases de données, Christian Soutou

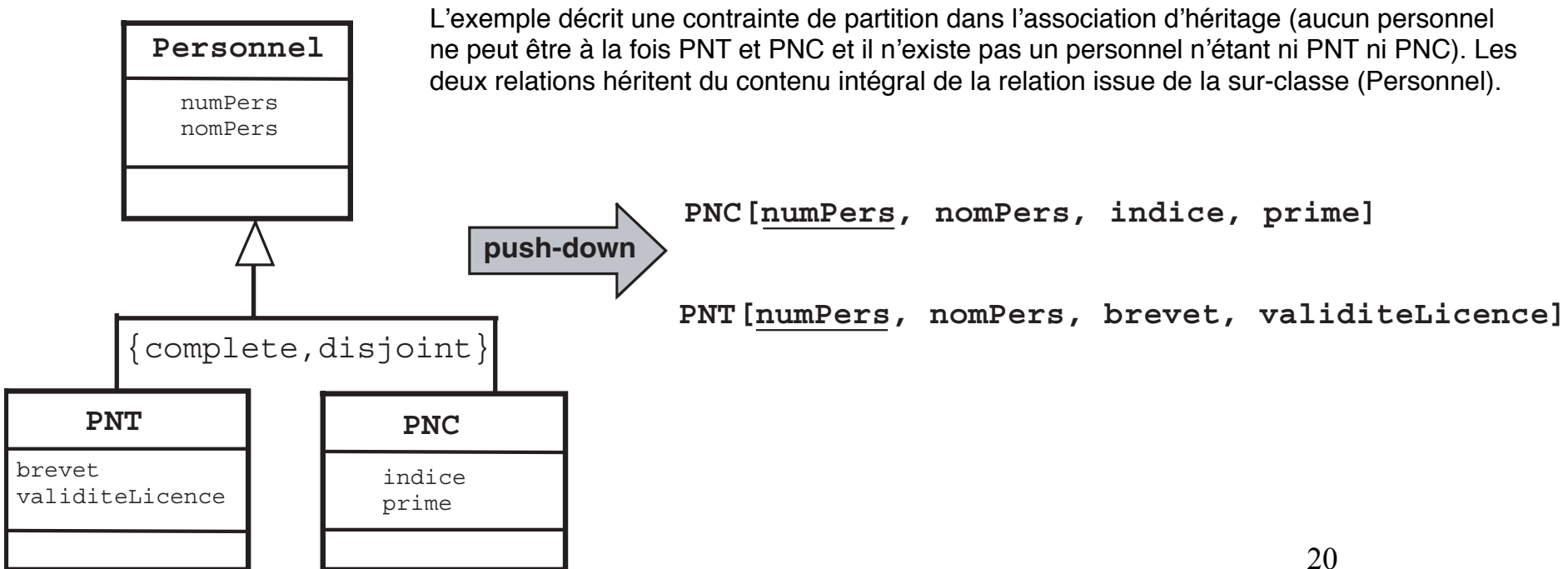
Décomposition par distinction

Chaque sous-classe est transformée en une relation.
La clé primaire de la sur-classe migre
dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et
devient à la fois clé primaire et clé étrangère.



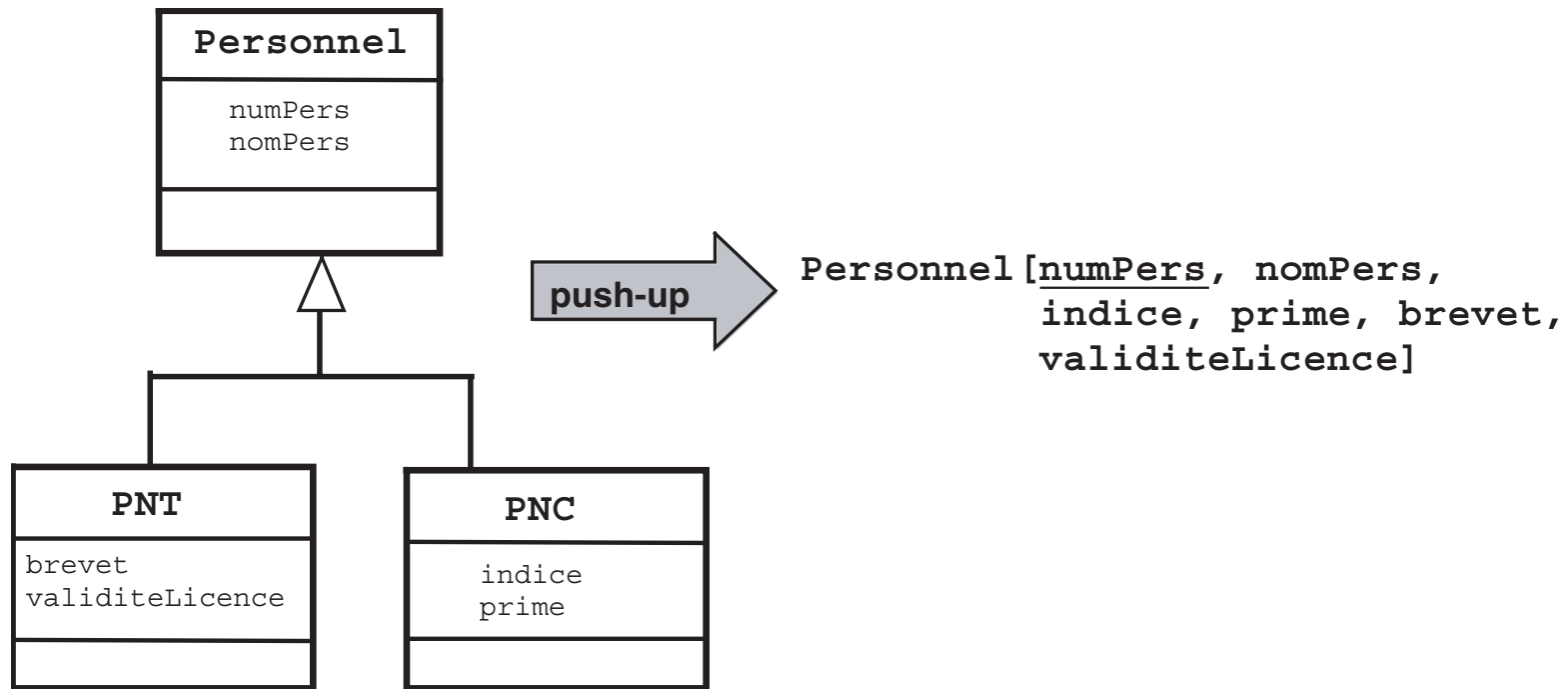
Décomposition descendante (push-down)

S'il existe une contrainte de totalité ou de partition sur l'association, il est possible de ne pas traduire la relation issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).



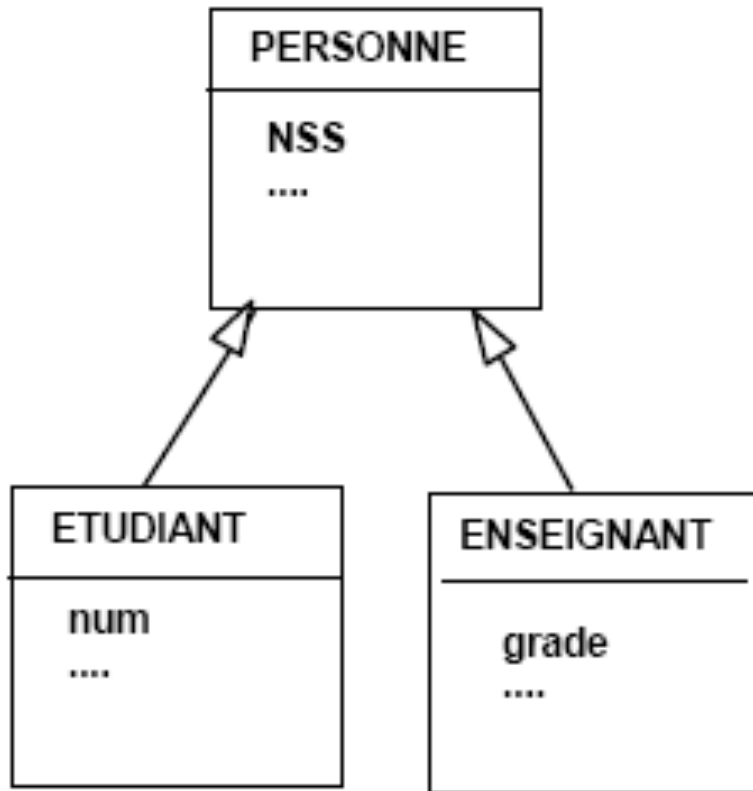
Décomposition ascendante (push-up)

Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la sur-classe.



Héritage

Spécialisation



Plusieurs choix

- « aplatir vers le haut »

Schéma Personne

Clé primaire

NSS

- « aplatir vers le bas »

Schémas Etudiant, Enseignant

Clé primaire

NSS

- **conserver les niveaux**

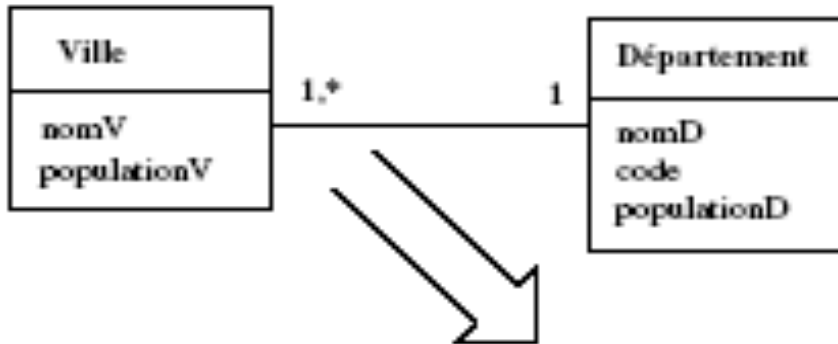
Sur les résultats associés aux séminaires?

Transformation des associations

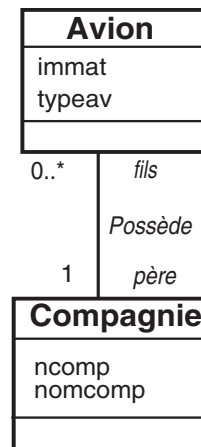
- Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations :
 - un-à-plusieurs ;
 - plusieurs-à-plusieurs ou classes-associations, et n-aires ;
 - un-à-un.

Association mono-multi

Ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.



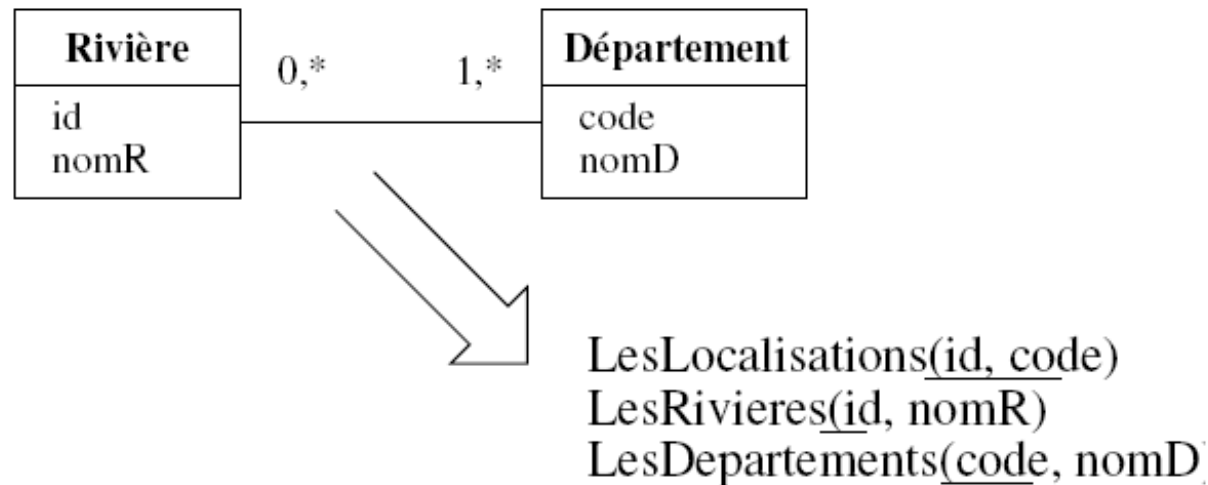
LesDepartements(code, nomD, populationD)
 LesVilles(nomV, populationV, code)



Avion [immat, typeav, ncomp#]
 Compagnie [ncomp, nomcomp]

Association multi-multi

L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des entités (classes) connectés à l'association.



Association mono-mono

Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à un.

Associations

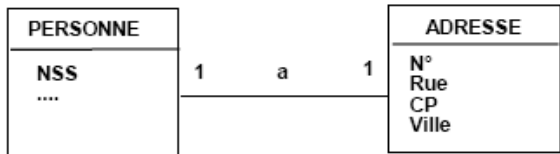
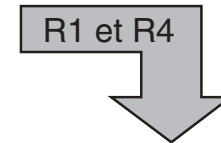
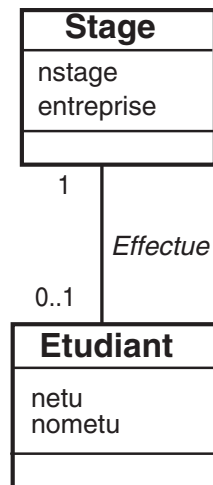
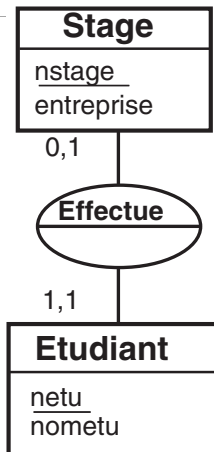


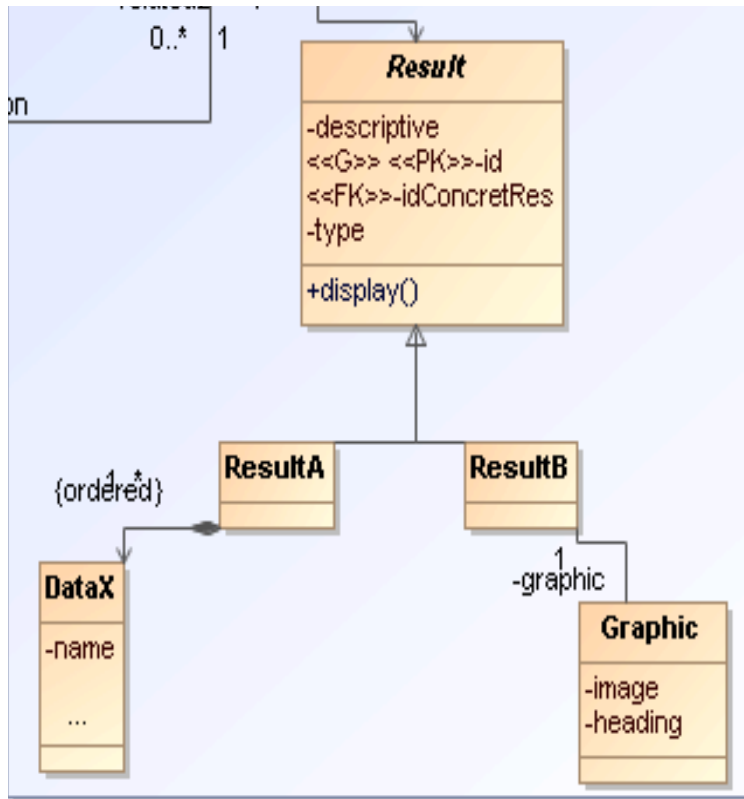
Schéma Personne

	Attribut	Domaine	Non Null
Ciè primaire NSS	<u>NSS</u>	String(13) ID	Oui
Ciè étrangère Id_adresse	Nom <u>Id_adresse</u>	String(35) Identifiant	Oui Oui



Stage [nstage, entreprise]
 Etudiant [netu, nometu, nstage#]

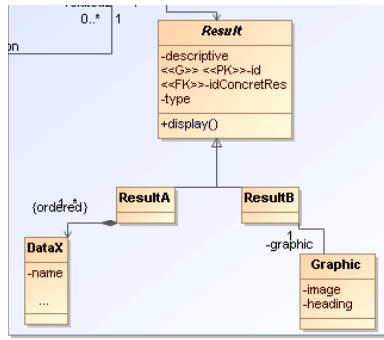
Héritage, Base de données et PHP



```
CREATE TABLE `Result` (  
  `descriptive` varchar(30) default NULL,  
  `id` int(11) NOT NULL auto_increment,  
  `idConcretResult` int(11) NOT NULL,  
  `type` varchar(8) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idConcretResult` (`idConcretResult`,`type`)  
)
```

```
CREATE TABLE `ResultB` (  
  `resultRef` int(11) NOT NULL,  
  `image` varchar(20) NOT NULL,  
  `id` int(11) NOT NULL auto_increment,  
  `heading` varchar(15) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `resultRef` (`resultRef`)  
)--
```

Héritage, Base de données et PHP

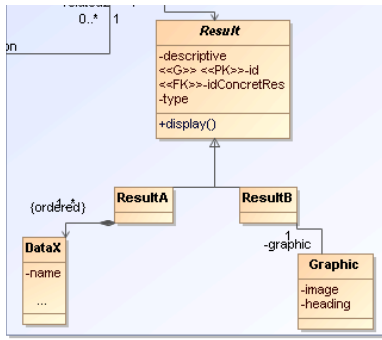


```
class Resultat {
    protected $_intitulé;
    private $_resultatConcret;
    protected $_table;

    public function __construct($intitule,$src="unknown",$t="inconnue")
    { $this->_intitulé = $intitule;
      $this->_resultatConcret = $src;
      $this->_table = $t; }

    public static function read($id)
    { Database::connect();
      $query = "SELECT * FROM resultat WHERE id = '". $id. "'";
      $ans = mysql_fetch_assoc(mysql_query($query));
      $i = $ans["intitulé"];
      $t = $ans["table"];
      $src = $ans["resultatConcret"];
      Database::disconnect();
      return new Resultat($i,$src,$t);
    }
}
```

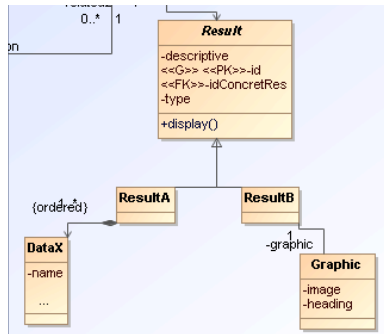
Héritage, Base de données et PHP



```
public function display()
```

```
{
    $res = $this->_intitulé;
    if ($this->_table == "inconnue") {
        return $res;
    }
    else {
        return $res . " est de type " . $this->_table ;
    }
}
```

Héritage, Base de données et PHP



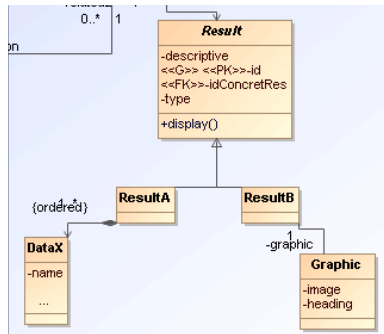
```
class ResultatAView extends Resultat
```

```
{
    private $_resultatRef;
    private $_procstockA;
    private $_contents;

    public function __construct($intitulé,$resultatRef,$procstockA="rien",
    $contents="rien")
    {
        $this->_intitulé=$intitulé; //par héritage pour php, jointure pour bd
        $this->_resultatRef = $resultatRef;
        $this->_procstockA = $procstockA;
        $this->_contents = $contents;
        parent::__construct("intit");
    }

    public static function read($id)
    {
        ...
        $query = "SELECT * FROM resultataview WHERE id = '". $id. "'";
        ...
        return new ResultatAView($intitule,$ref,$pa,$c);
    }
}
```

Héritage, Base de données et PHP

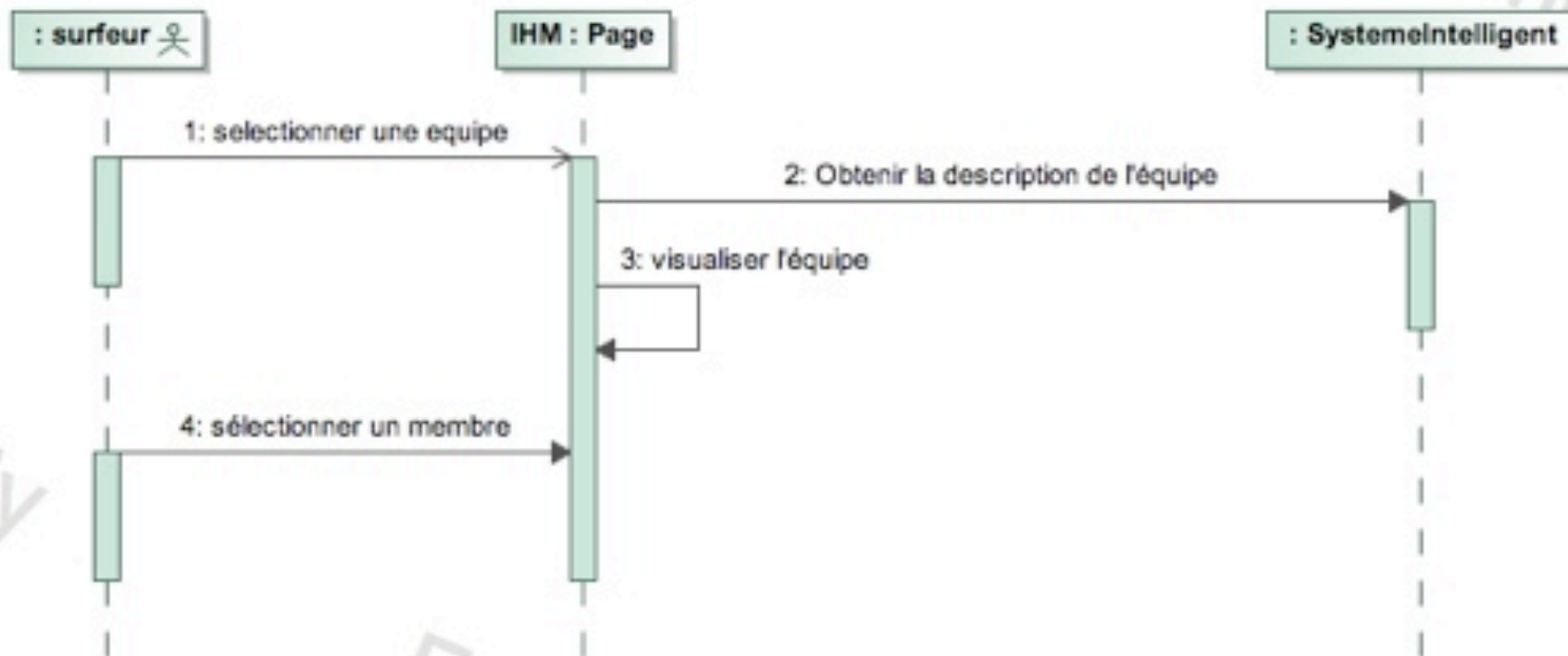


Utilisation

```
function getResult($id)
{
    Database::connect();
    $query = "SELECT * FROM resultat WHERE id LIKE '". $id.'";
    $ans = mysql_fetch_assoc(mysql_query($query));
    $i = $ans["intitulé"];
    $t = $ans["table"];
    $src = $ans["resultatConcret"];
    if ($t == "resultata") {
        $tmpRes = ResultatAView::read($src);
    }
    else if ...
    else $tmpRes = Resultat::read($id);

    Database::disconnect();
    return $tmpRes;
}
```


Retour sur les Scénarios



Scenari : Vers l'architecture logicielle

