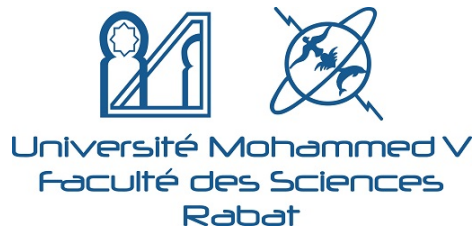


UNIVERSITÉ MOHAMMED V de Rabat
Faculté des Sciences



Département d'Informatique

Master Spécialisé en Informatique Appliqué Offshoring

Réalisation d'un processus métier sous
bonita

Présenté par :
AIT LHOUARI OUSSAMA
BOUDOUAR MOHAMED

Encadré par :
DR. SAID SLAOUI

Année universitaire 2018-2019

Table des matières

Introduction	4
1 Environnement technique et outils	5
1.1 Environnement de développement	5
1.1.1 JDK : Java Development Kit	5
1.1.2 Bonita	5
1.1.3 Eclipse	6
1.1.4 Apache Tomcat	6
1.1.5 Apache Axis2	7
1.1.6 WampServer	7
1.1.7 SoapUI	8
1.2 Terminologie	8
1.2.1 Web Service	8
1.2.2 WSDL	9
1.2.3 UDDI	10
1.2.4 workflow	10
1.2.5 Moteur de workflow	10
2 Étude de cas et conception	11
2.1 Énoncé du projet	11
2.2 Étude conceptuelle	11
2.2.1 Diagramme des classes	12
2.2.2 Diagramme des données	13
3 La réalisation du projet	15
3.1 Réalisation des services web	15
3.2 création des processus métiers sous bonita	17
3.2.1 Définition du modèle de données métier	17
3.2.2 déclaration des variables du pool	19
3.2.3 Définition des contrats	19
3.2.4 Création des formulaires	20
3.2.5 définition des portes logique	21
3.2.6 Création des connecteurs en entrée	21

4	Démonstration	26
4.1	Processus de paiement échelonnée	26
4.1.1	la demande de paiement échelonné acceptée	29
4.1.2	Le cas d'un mauvais payeur	30
4.2	Paiement des échéances	32
	Conclusion	36

Introduction

L'accès aux systèmes d'information s'appuie aujourd'hui de plus en plus sur des technologies Internet. Les efforts de standardisation dans ce contexte ont accentué l'engouement des personnes et des organisations pour l'utilisation de l'Internet et ont permis l'émergence des services Web comme support de développement des applications accessibles par Internet. Ainsi, les technologies associées aux services Web sont devenues incontournables pour le développement d'applications interagissant les unes avec les autres par le biais de l'Internet.

Les Web Services sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standards. Les Web Services définissent non seulement les données transmises entre deux applications, mais aussi comment traiter ces données et les relier à l'interne et à l'externe d'une application logicielle sous-jacente. De ce fait, les Web Services permettent aux entreprises et individus de publier des liens vers leurs applications de la même manière qu'ils publient des liens vers leurs pages Web. Conséquemment, les Web Services peuvent faire en sorte que toutes les ressources informatiques dont une entreprise a besoin soient des ressources distribuées à la grandeur de l'Internet.

Le but de ce projet est d'implémenter le processus paiement échelonné de l'étude de cas du tour opérateur en utilisant les web services sous JEE et Bonita comme moteur BPM.

Dans ce projet, on a réalisé une application basée sur des services web et sur un client BPM. Une application qui repose sur la technologie Web Service SOAP. On présente donc une synthèse de ce travail dans ce rapport. Commencant par une présentation contextuelle de l'application et de sa conception. Puis la mise en œuvre de différents outils et technologies utilisés. Et après, une illustration des étapes de la réalisation. Avant de conclure par une démonstration d'un exemple d'utilisation. **Mots clés :** BPM ; BONITA ; WEB services ; Java EE ; SOAP ; MySQL ;

Chapitre 1

Environnement technique et outils

Dans ce chapitre on présente l'environnement et les outils utilisé pour réaliser ce projet.

1.1 Environnement de développement

1.1.1 JDK : Java Development Kit

Le JDK (Java Development Kit en anglais, Kit de Développement Java en Français) représente l'outillage indispensable au développeur Java. Ce kit contient les outils nécessaires pour programmer en java, exécuter ses programmes java, les tester et les livrer à ses clients.



FIGURE 1.1.1 – Java Development Kit

1.1.2 Bonita

Bonita BPM est une solution open source pour la gestion de processus métiers. C'est un moyen de définir et de gérer les étapes d'un processus métier, du début jusqu'à la fin. Ces processus consistent en une séquence d'activités présentant un intérêt pour une organisation.



FIGURE 1.1.2 – Bonita

1.1.3 Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Figurant parmi les grandes réussites de l'open source, Eclipse est devenu un standard du marché des logiciels de développement.



FIGURE 1.1.3 – Eclipse

1.1.4 Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process⁶, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP. Tomcat est un serveur HTTP à part entière⁷. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant. la version utilisé pour réaliser ce projet est 8.0.53.



FIGURE 1.1.4 – Apache Tomcat

1.1.5 Apache Axis2

Apache Axis2 est un moteur de base pour les services Web. Il s'agit d'une refonte complète de la pile SOAP Apache Axis largement utilisée. Les implémentations d'Axis2 sont disponibles en Java et en C. Axis2 permet d'ajouter des interfaces de services Web aux applications Web. Il peut également fonctionner en tant que serveur d'applications autonome.



FIGURE 1.1.5 – Apache Axis2

1.1.6 WampServer

WampServer est une plate-forme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.



FIGURE 1.1.6 – WampServer

1.1.7 SoapUI

SoapUI est une application open source permettant le test de web service dans une architecture orientée services (SOA). Ses fonctionnalités incluent l'inspection des web service, l'invocation, le développement, la simulation, le mocking, les tests fonctionnels, les tests de charge et de conformité.



FIGURE 1.1.7 – SoapUI

1.2 Terminologie

1.2.1 Web Service

Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel. **Architecture des Web Service**

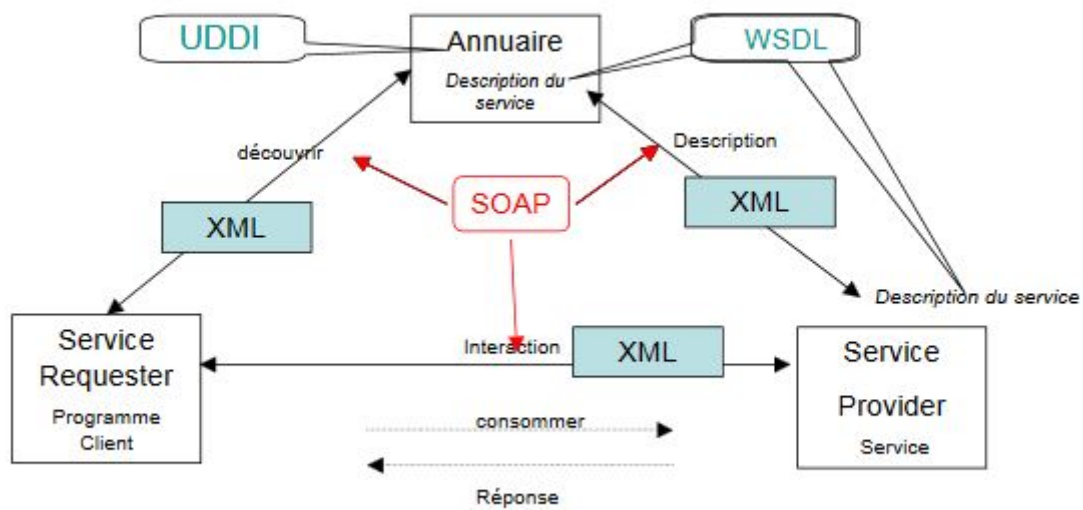


FIGURE 1.2.1 – Architecture des Web Services

Les acteurs dans un web service

- **Annuaire :**
 - Annuaire des services publiés par les providers (UDDI).
 - Géré sur un serveur niveau application, entreprise ou mondial
- **Service Provider**
Application s'exécutant sur un serveur et comportant un module logiciel accessible par en XML.
- **Service Requester**
Application cliente se liant à un service et invoquant ses fonctions par des messages XML (REST, XML-RPC, SOAP).

1.2.2 WSDL

Le standard WSDL (Web Service Description Language) est un langage reposant sur la notation XML permettant de décrire les services web. WSDL permet ainsi de décrire l'emplacement du service web ainsi que les opérations (méthodes, paramètres et valeurs de retour) que le service propose. **Structure d'un document WSDL :**

- **PortType :** définissant le service web, en particulier les opérations qu'il réalise et le type de messages échangés.
- **Message :** comprenant une ou plusieurs parties représentant les paramètres d'entrée.
- **Types :** Définissant les types de données utilisés par le service web.
- **Binding :** Précisant le protocole utilisé et le format de message.

1.2.3 UDDI

UDDI (Universal Description Discovery and Integration) est un standard adopté dans sa version 2 par l'organisme OASIS (Organization for the Advancement of Structured Information Standards) en 2001.

Le but de ce standard est de fournir des fonctionnalités d'annuaires des WebServices afin de pouvoir les découvrir et les utiliser via un mécanisme centralisé pouvant regrouper les WebServices de plusieurs entreprises. Il repose sur SOAP pour sa partie d'échange de messages, et sur WSDL pour sa partie de description des services de l'annuaire.

1.2.4 workflow

On appelle "Workflow" la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un **processus métier**. Le terme de Workflow pourrait donc être traduit en français par **Gestion électronique des processus métier**.

Le workflow permet aux entreprises d'automatiser les processus, procédures et tâches de l'entreprise d'un participant à un autre, grâce à l'utilisation de l'informatique, des réseaux intranet et Internet, de bases de données, du courrier électronique, des logiciels bureautiques

1.2.5 Moteur de workflow

Le moteur de workflow est l'outil permettant de modéliser et d'automatiser les processus métiers de l'entreprise. Ce type d'outil permet ainsi de formaliser les règles métier de l'entreprise afin d'automatiser la prise de décision, c'est-à-dire la branche du workflow à choisir, en fonction du contexte donné.

Bonita BPM est un moteur de modélisation de processus, ce dernier est utilisé pour réaliser ce projet.

Chapitre 2

Étude de cas et conception

2.1 Énoncé du projet

Les clients du tour opérateur ont le choix entre deux type de paiement, soit paiement direct en payant la totalité du tarif du voyage sélectionné, ou en demandant un paiement échelonné, ce dernier est le sujet de notre projet, quand le client demande ce type de paiement l'agent commercial doit obtenir des informations du client afin de constituer son dossier de paiement échelonné, ces information sont l'adresse, le métier et le salaire mensuel. après constitution du dossier il est envoyé vers la direction financière pour l'étudier et donner une réponse soit au client directement ou à l'agence du voyage dans un délai de moins de huit jours. Pour ce paiement il existe sept échéance, l'acompte est la première puis six autres échéance chaque mois, à partir du mois suivant la fin des vacances. le SI cible doit permettre aux agents commerciaux de donner une réponse immédiate pour les clients qui demandent un paiement échelonné, aussi de payer leurs échéance de la manière souhaiter par eux en respectant le délai de six mois qui ne changera pas avec le nouveau SI. Le client se présente dans l'agence du voyage, sélectionne son voyage, les activités et le type de logement, puis l'agent commercial lui donne le tarif du voyage, le client fait le choix entre les deux paiement, et il paye l'acompte qui est d'au moins 10% du tarif du voyage. l'agent commercial saisit les données du client et du voyage choisis, consulte la liste noir pour vérifier si le client est éligible pour avoir un paiement échelonné, si oui la réservation est définitive, sinon l'agent demande au client soit de payer la totalité du tarif du voyage immédiatement, ou de le rembourser. après le paiement de chaque échéance le client peut demander une facture, voir la somme qu'il a payé et le reste à payer.

2.2 Étude conceptuelle

Après l'analyse de l'étude de cas du tour opérateur on a pu déterminer les entités qui doivent être liée entre eux afin de réaliser le processus métier.

Client : Pour les commerces modernes, le client constitue le noyau de toute action, bien évidemment de tout prot qui vient avec, les sociétés de nos jours, font de leurs bases les relations avec leur clientèle, et à chaque moment ces relations améliorent en terme de fidélisation et diversité des services, les profits de ces dernières augmentent. Dans notre cas, c'est le client qui se dirige vers une agence pour pouvoir effectuer une réservation,

bien entendu, après avoir vu une offre par le biais de la publicité, ou bien qu'il l'a vue sur le catalogue du tour opérateur.

Voyage : le voyage pour le tour opérateur constitue le produit, bien sûr se figurant comme un service non tangible, mais désormais on a trouvé comment le faire tendre pour devenir un produit vendable. Pour le processus de réservation, l'accomplissement souhaité c'est de pouvoir satisfaire le client en lui garantissant un voyage, selon des conditions de disponibilité de ce dernier, et aussi de la solvabilité du client qui souhaite le commander.

Paiement : Après avoir choisi un voyage par un certain client, la troisième partie de l'architecture commerciale doit être prise en compte, c'est le processus du paiement. Pour le tour opérateur, le paiement se fait avec de nombreuses possibilités :

- **Paiement échelonné :** c'est une procédure qui permet au client de ne pas verser la somme dont il est redevable pour un voyage d'un seul coup, d'abord il paie une somme que l'on appelle l'acompte, en suite on lui réserve son voyage désiré, en tenant compte qu'après son retour, il sera encore redevable de ce qui reste de la somme totale du voyage dont il a bénéficié.
- **Paiement direct :** le client dans ce cas paie la somme de son voyage.

Facture : c'est l'entité responsable sur toutes les transactions, toutes les réservations sont liées à une facture, les clients aussi sont liés à cette entité.

Paiement : cette entité est liée à la facture, toutes les transactions sont liées à cette entité. tout paiement est lié à une seule facture.

2.2.1 Diagramme des classes

Le diagramme de classes donne une représentation des objets du système qui vont interagir ensemble pour permettre aux processus métier d'être exécutés.

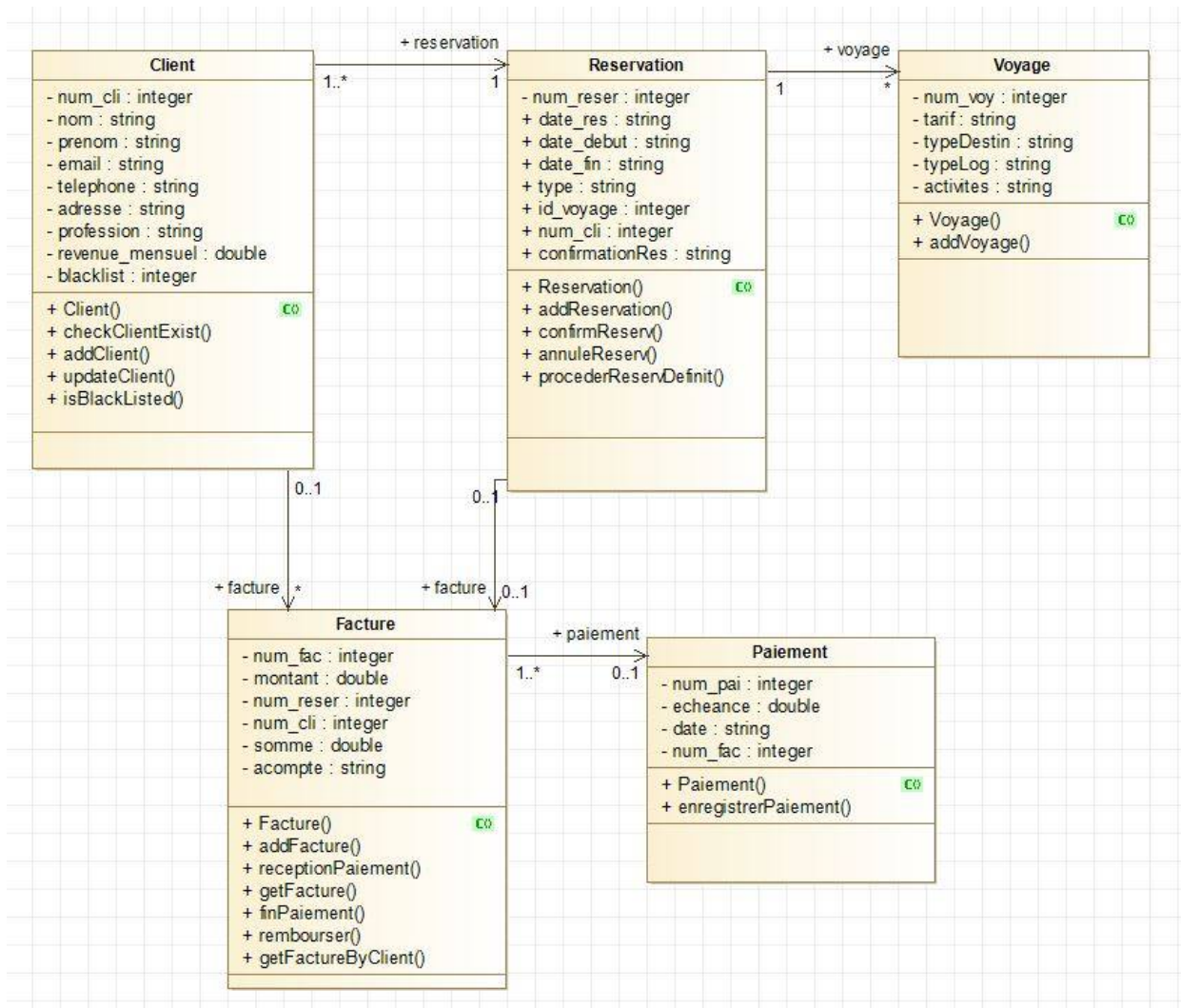


FIGURE 2.2.1 – diagramme des classes

2.2.2 Diagramme des données

Les tables de notre base de données contiennent les attributs de chaque classe du diagramme de classe, avec les mêmes liens entre classes.

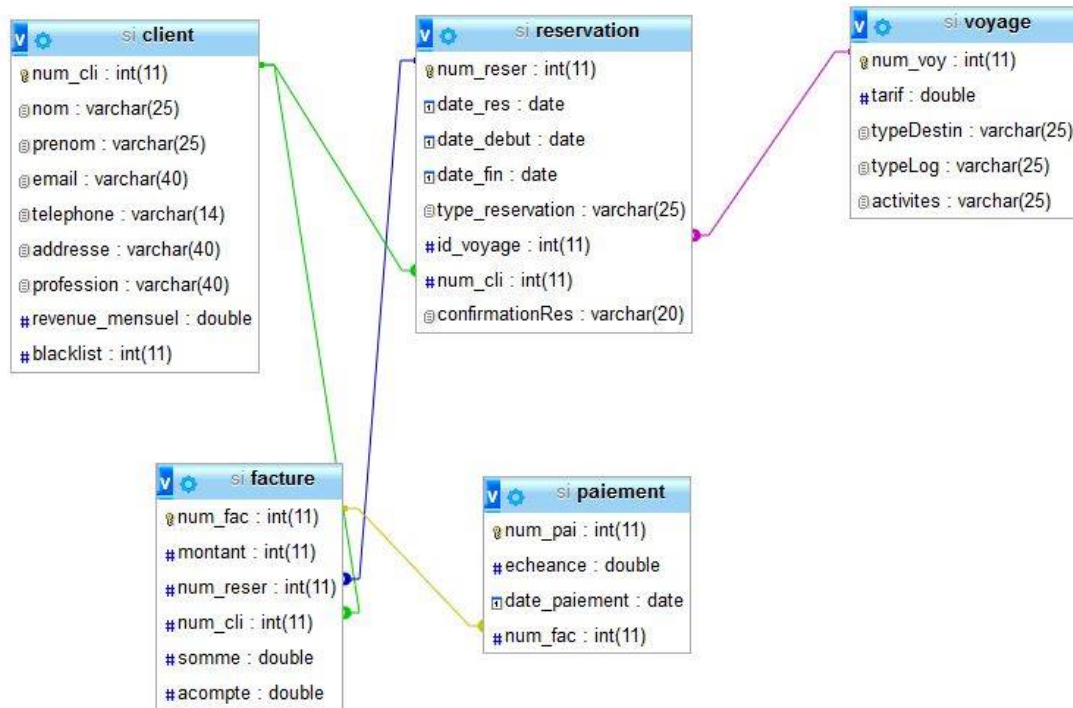


FIGURE 2.2.2 – diagramme des données

Chapitre 3

La réalisation du projet

3.1 Réalisation des services web

Après création des classes et leurs méthodes, on a crée une classe nommé "**WebServices**" qui va être publié comme web service afin d'être le lien entre bonita et méthodes d'accès à la base de données. cette classe fournit les fonctions suivante :



FIGURE 3.1.1 – méthodes de la classes WebServices

la création d'un web service publiant les méthodes de la classes **WebServices**.

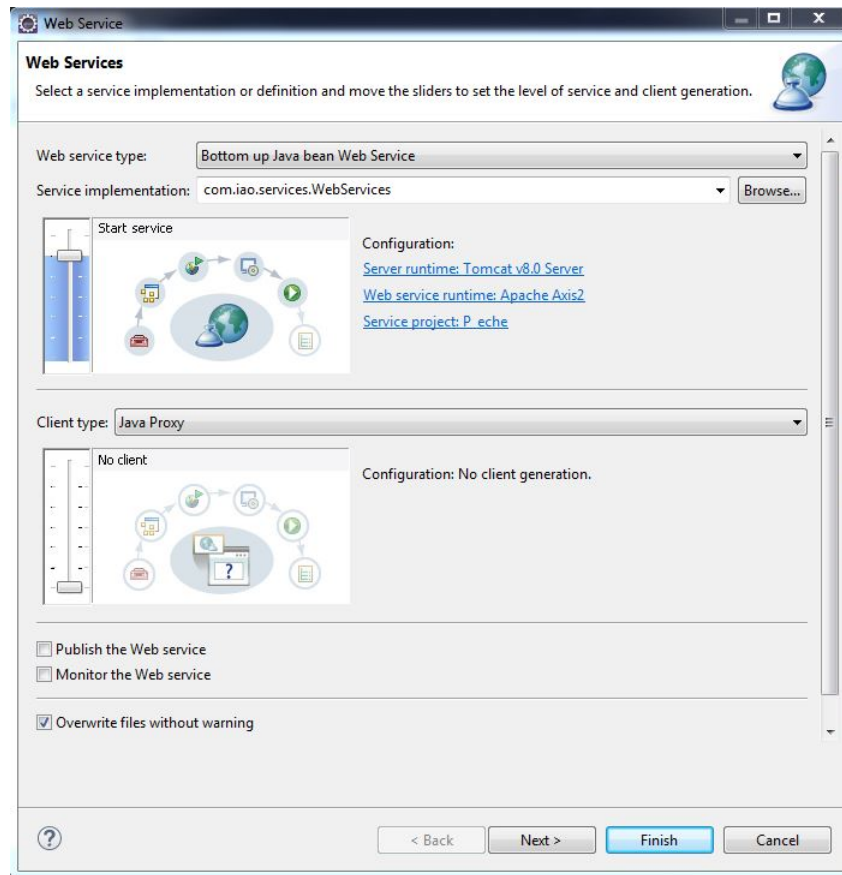


FIGURE 3.1.2 – Création du service web

les méthodes publiées par le web service qu'on a créé :

Available services

WebServices

Service Description : Please Type your service description here

Service EPR : http://localhost:8080/P_eche/services/WebServices

Service Status : Active

Available Operations

- annuleReserv
- preparationDossier
- payeecheance
- confirmPreres
- etudeDossier
- paiementImmediat
- procederReservDefinit
- finPaiement
- creeReservation
- paiementImmediatClient
- rembourser
- getFactureByClient
- getFacture
- creeClient

FIGURE 3.1.3 – méthodes publiées par le web service

3.2 création des processus métiers sous bonita

Après la génération du web service, on passe à la deuxième étape de réalisation du projet, qui consiste à concevoir le workflow sous bonita 7.8.0

Le processus ci-dessous décrit les différentes tâches modélisées sous Bonita.

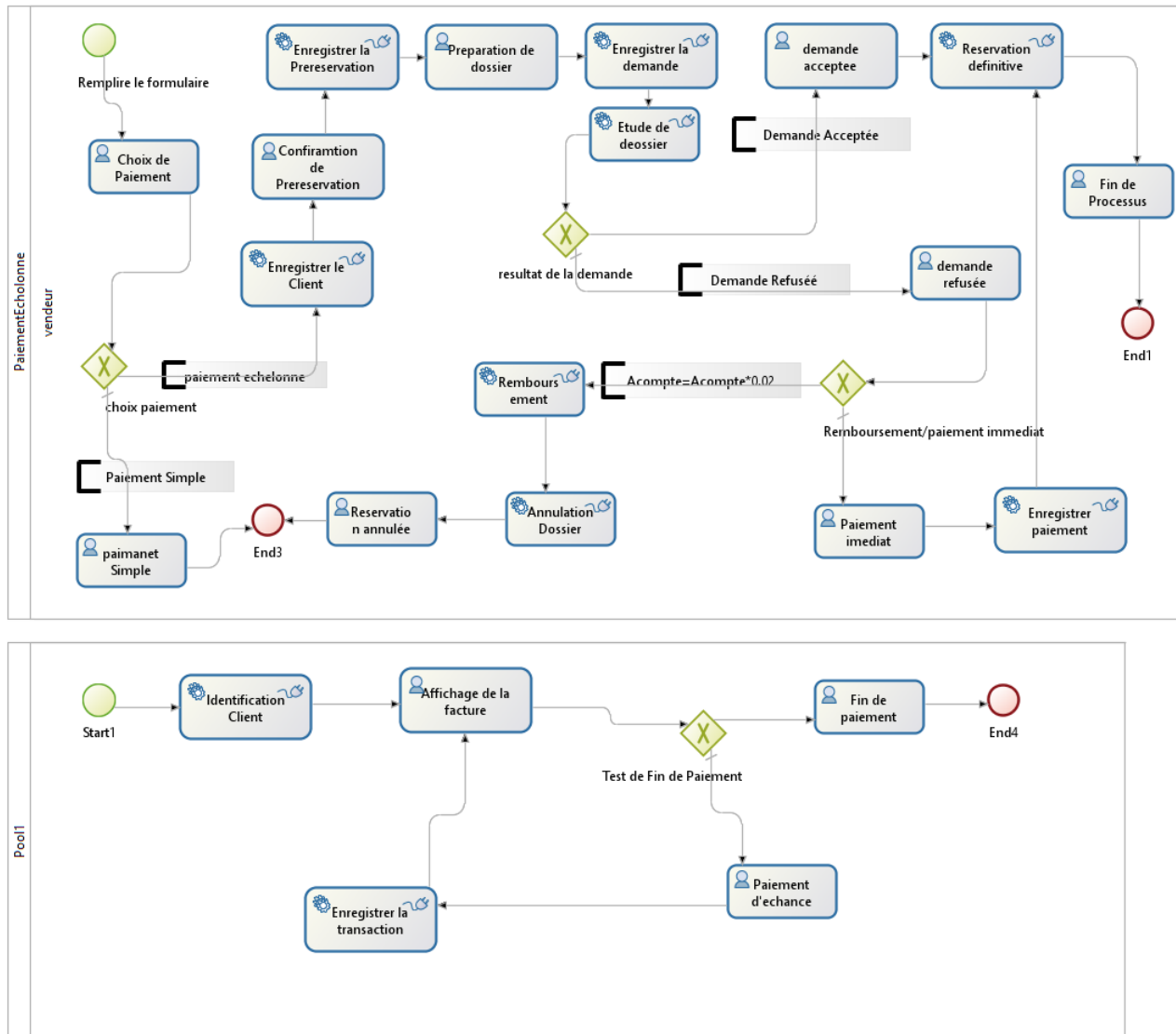


FIGURE 3.2.1 – Processus metiers

3.2.1 Définition du modèle de données métier

La première étape dans notre réalisation de notre workflow sous bonita consiste à définir le modèle de données métier.

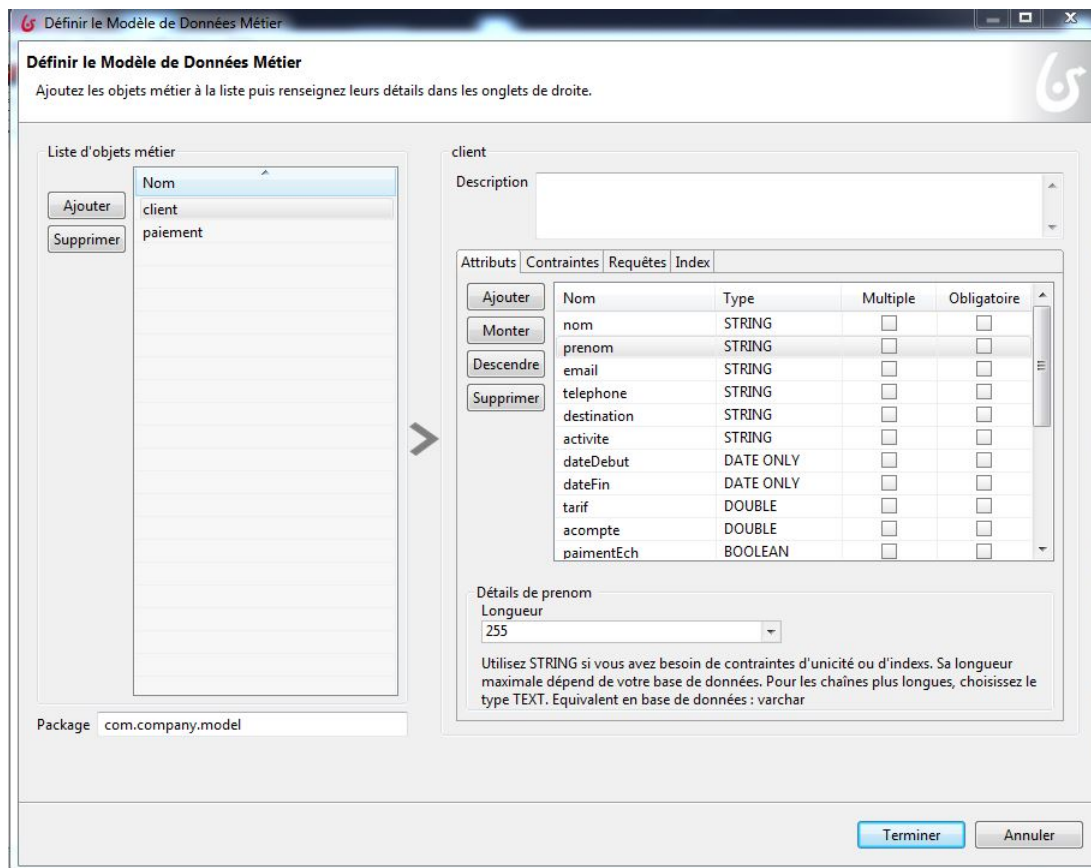


FIGURE 3.2.2 – Définition du modèle de données métier

dans notre projet on a crée deux objets métier, le premier est l'objet Client, le deuxième est l'objet paiement. le premier objet est utilisé dans le premier pool, le deuxième est utilisé dans le deuxième pool.

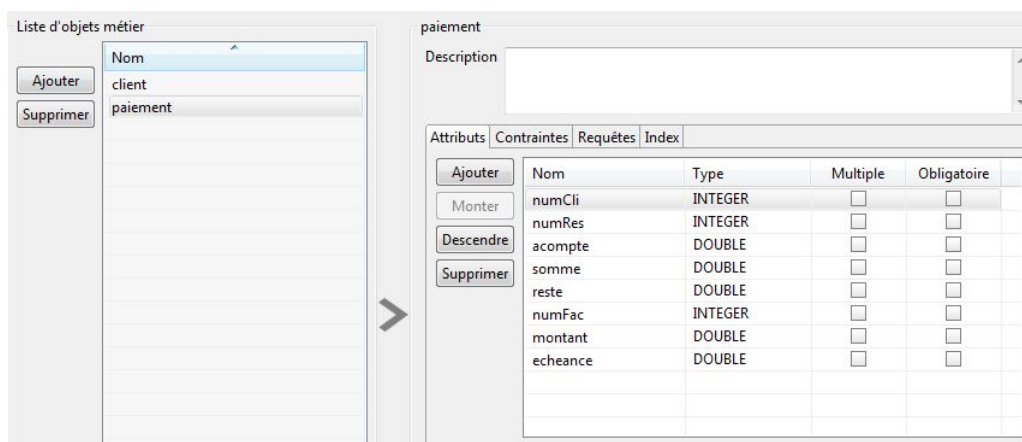


FIGURE 3.2.3 – Définition de l'objet métier paiement

3.2.2 déclaration des variables du pool

La deuxième étape dans notre réalisation consiste à déclarer les variables de chaque pool en indiquant leurs valeurs par défaut.

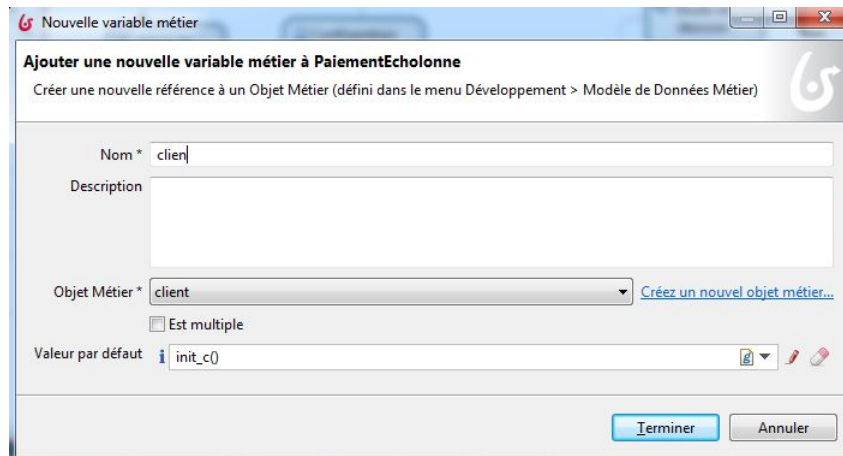


FIGURE 3.2.4 – Déclaration de la variable client dans le pool

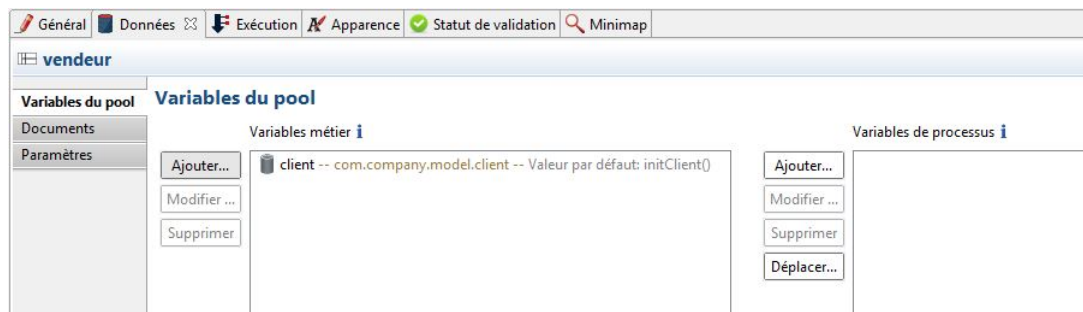


FIGURE 3.2.5 – Déclaration de la variable client dans le pool

3.2.3 Définition des contrats

Dans chaque tâche humaine ou automatique on doit définir le contrat, afin de garantir l'obtention des résultats souhaités après l'exécution de cette tâche. Le premier contrat à définir est celui qui doit être respecté après remplir le premier formulaire, consiste à récupérer l'objet **ClienInput** contenant les données du client, et du voyage qu'il veut réserver.

Il est conseillé de d'abord définir des [variables métier](#) et/ou des [documents](#) puis de cliquer sur « Ajouter à partir de données... ».

Cette fonction créera automatiquement le mapping entre inputs de contrat et données. Elle initialisera aussi les données avec les valeurs du contrat.

Nom *	Type	Multiple	Description
clientInput	COMPLEX	<input type="checkbox"/>	
nom	TEXT	<input type="checkbox"/>	
prenom	TEXT	<input type="checkbox"/>	
email	TEXT	<input type="checkbox"/>	
telephone	TEXT	<input type="checkbox"/>	
destination	TEXT	<input type="checkbox"/>	
activite	TEXT	<input type="checkbox"/>	
dateDebut	DATE ONLY	<input type="checkbox"/>	
dateFin	DATE ONLY	<input type="checkbox"/>	
tarif	DECIMAL	<input type="checkbox"/>	
acompte	DECIMAL	<input type="checkbox"/>	

FIGURE 3.2.6 – définition du contrat

3.2.4 Création des formulaires

La création des formulaires est l'une des importante tache, car l'interface doit être simple à utiliser, contenant les champs nécessaire pour respecter les contrats définit pour chaque tache.

Formulaire d'instanciation

Sélectionnez le type de formulaire utilisé pour instancier le processus. en édition Souscription

☒ UI Designer
 ☐ URL externe
 ☐ Aucun formulaire

Formulaire cible: FormStart

FIGURE 3.2.7 – Formulaire d'instanciation

le formulaire doit retourner un objet JSON contenant clientInput afin de respecter le contrat définit avant.

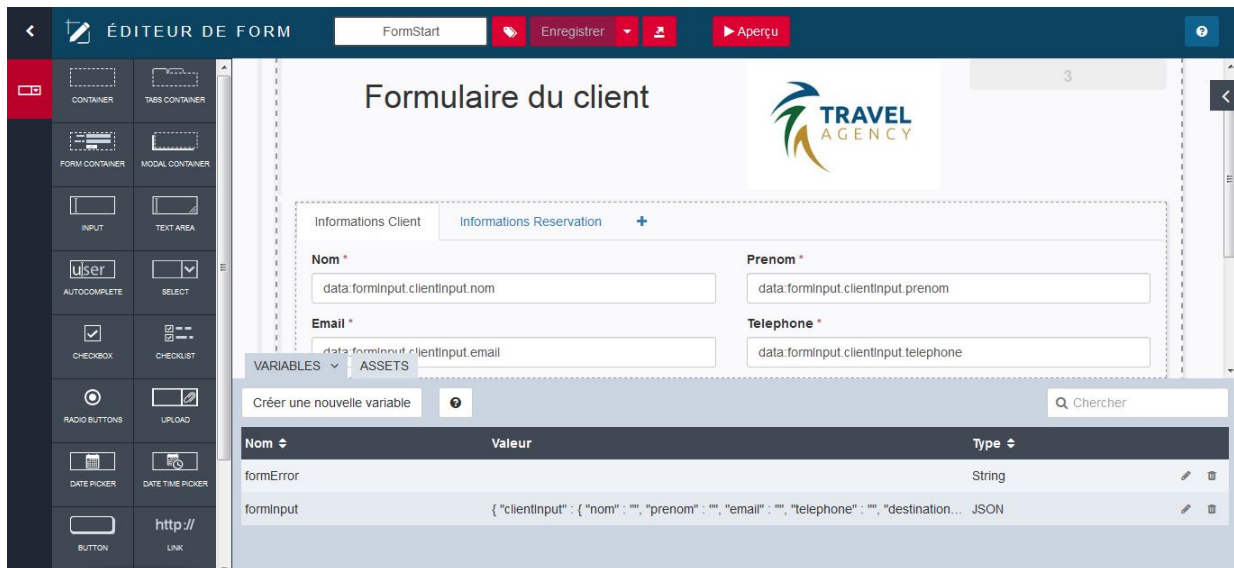


FIGURE 3.2.8 – Formulaire d’instanciation

3.2.5 définition des portes logique

Pour définir quel transition suivre après une tache, on définit la condition pour suivre une transition, et pour l’autre transition on sélectionne le flux par défaut. la première porte logique exclusive de notre workflow est situé après le choix du type de paiement, elle est suivi de deux transition, une vers l’enregistrement du client si le client a choisi le paiement échelonné, l’autre vers le paiement simple. on définit la condition de transition vers la tache **Enregistrer le client** :

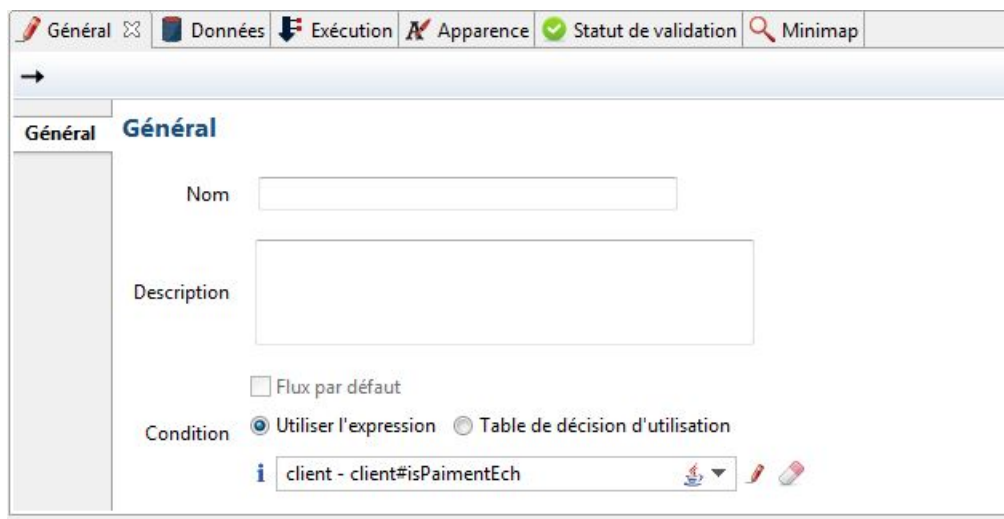


FIGURE 3.2.9 – Condition de transition

3.2.6 Création des connecteurs en entrée

Les connecteurs en entrée sont le lien entre bonita et les web services qu’on a crée. pour chaque tache automatique on doit crée des connecteurs en entrée afin d’exécuter ses

taches.

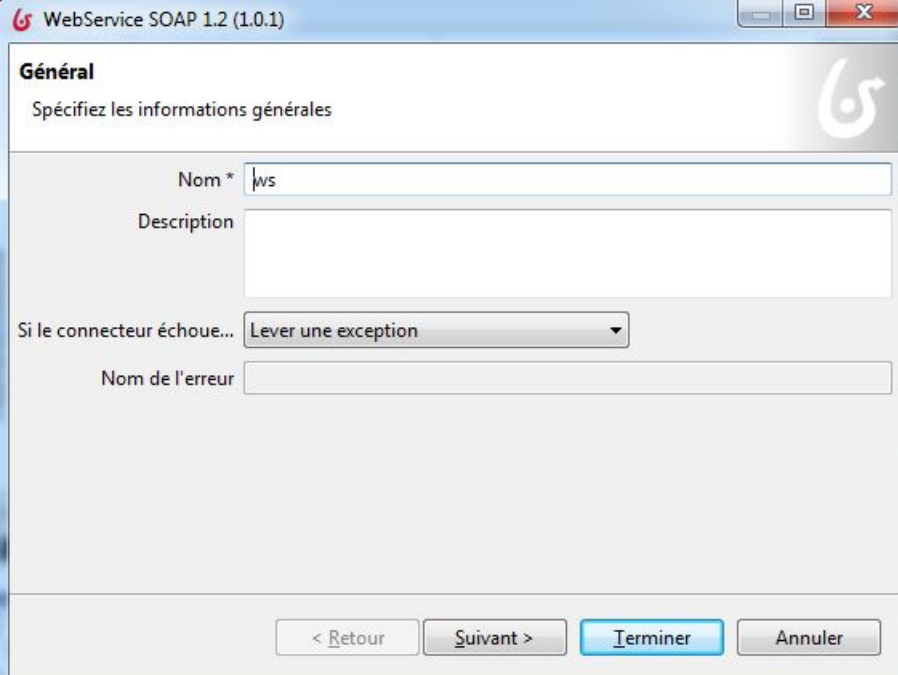


FIGURE 3.2.10 – Création d'un connecteur en entrée étape1

NS service : ce champ contient la valeur du targetNamespace du WSDL.
Nom du service : ce champ contient la valeur de l'attribut name de la balise WSDL :service du fichier WSDL.

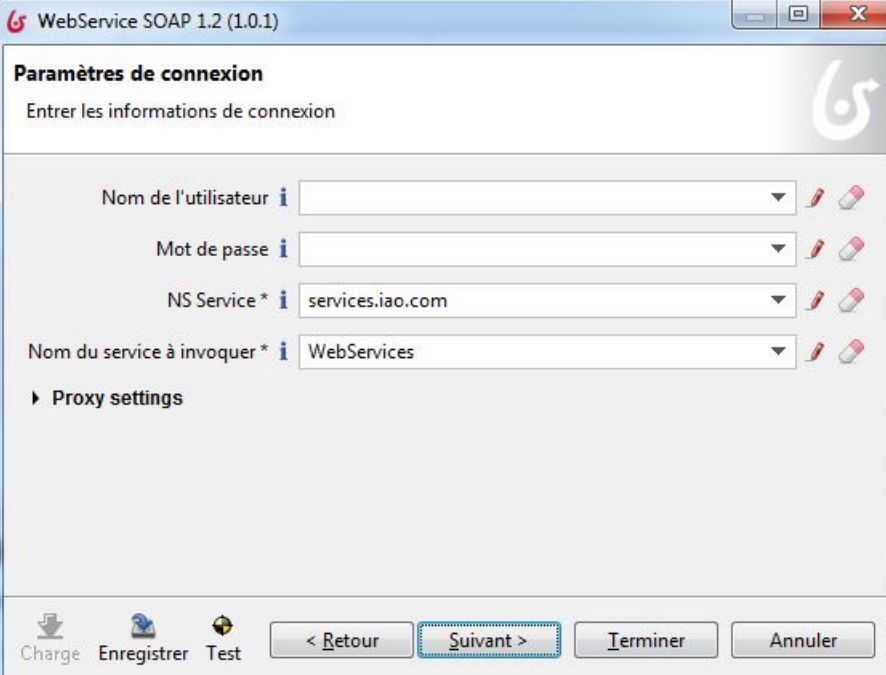


FIGURE 3.2.11 – Création d'un connecteur en entrée étape2

Nom du port : ce champ contient la valeur de l'attribut name de la balise "WSDL :port" du fichier WSDL.

Adresse du point d'encrage : ce champ contient la valeur de l'attribut location de la balise "soap :address".

Mode d'accès : `http://www.w3.org/2003/05/soap/bindings/HTTP/`

enveloppe : c'est là où on définit l'enveloppe SOAP.

WebService SOAP 1.2 (1.0.1)

Paramètres de la requête
Renseigner les informations relatives au Webservice

Action SOAP *i*

Nom du port * *i*

Adresse du point d'encrage * *i*

Mode d'accès * *i*

Enveloppe * *i* [Changer d'éditeur...](#)

FIGURE 3.2.12 – Création d'un connecteur en entrée étape3

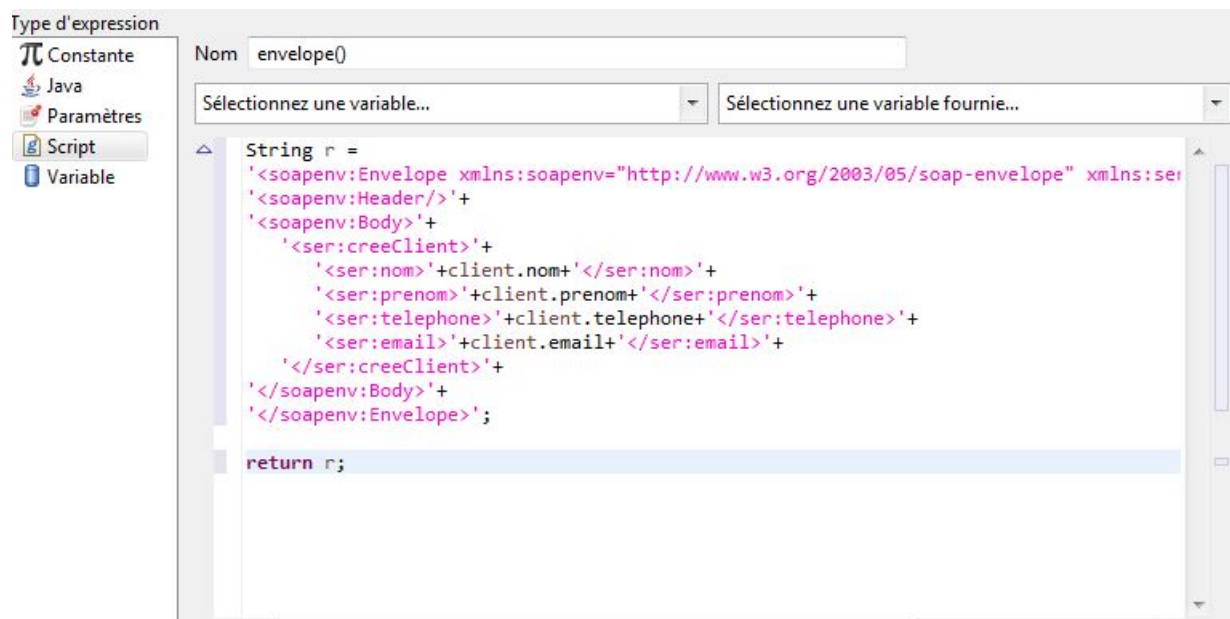


FIGURE 3.2.13 – Création de l’enveloppe SOAP

Après l’exécution de la méthode invoqué par bonita, on doit définir la méthode de réception des réponse reçu par les web service.

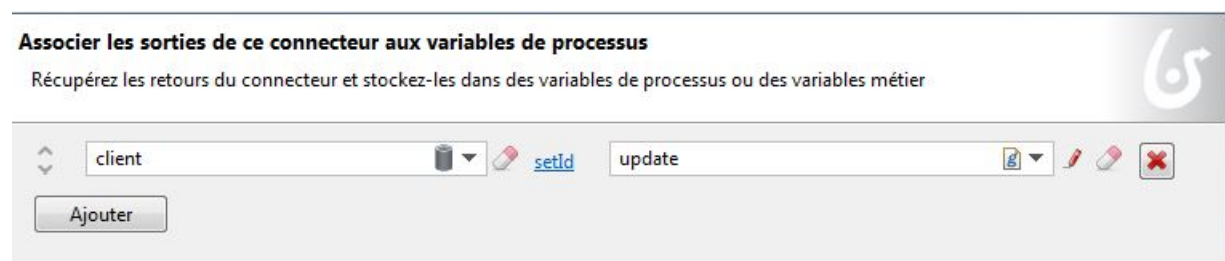


FIGURE 3.2.14 – Association des sorties du connecteur aux variables de processus

dans ce cas on reçoit l’id du client après sont ajout dans la base de données, on doit mettre à jour la variable id de l’objet client déclaré dans notre pool. on fait appel à setId qui va affecter la valeur qu’on a extrait de la reponse SOAP à l’id de la variable client.

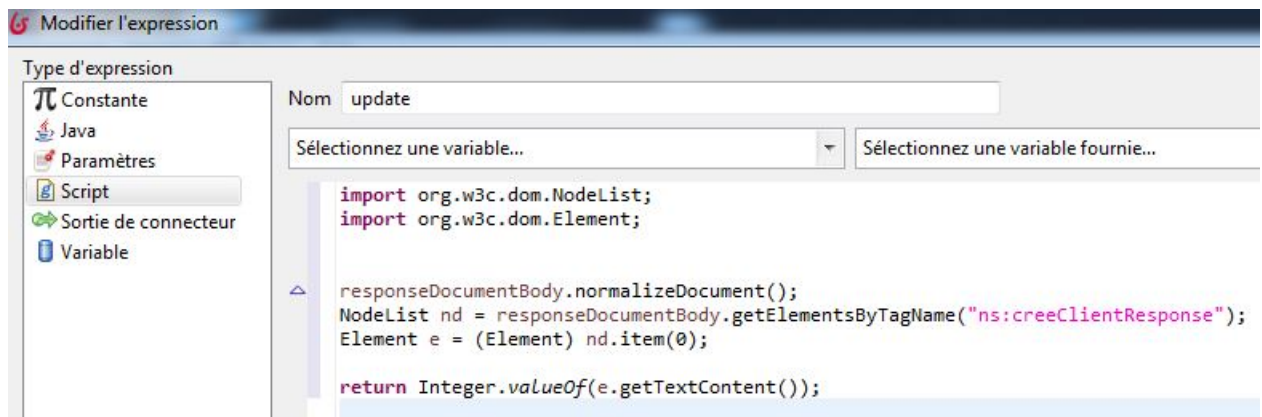


FIGURE 3.2.15 – Extraction de l'id à partir de la reponse SOAP

Chapitre 4


Démonstration

Dans ce chapitre on vas démontrer les différent processus métiers Créer et les différentes cas.

4.1 Processus de paiement échelonnée

Premièrement le vendeur saisie les informations d'un client et son choix de réservation

Formulaire du client



Informations Client

Informations Reservation

Nom*

Boudouar

Email*

mo@gmail.com

Prenom*

Mohamed

Telephone*

0600476989

Valider

FIGURE 4.1.1 – Saisir les informations d'un client

Formulaire du client



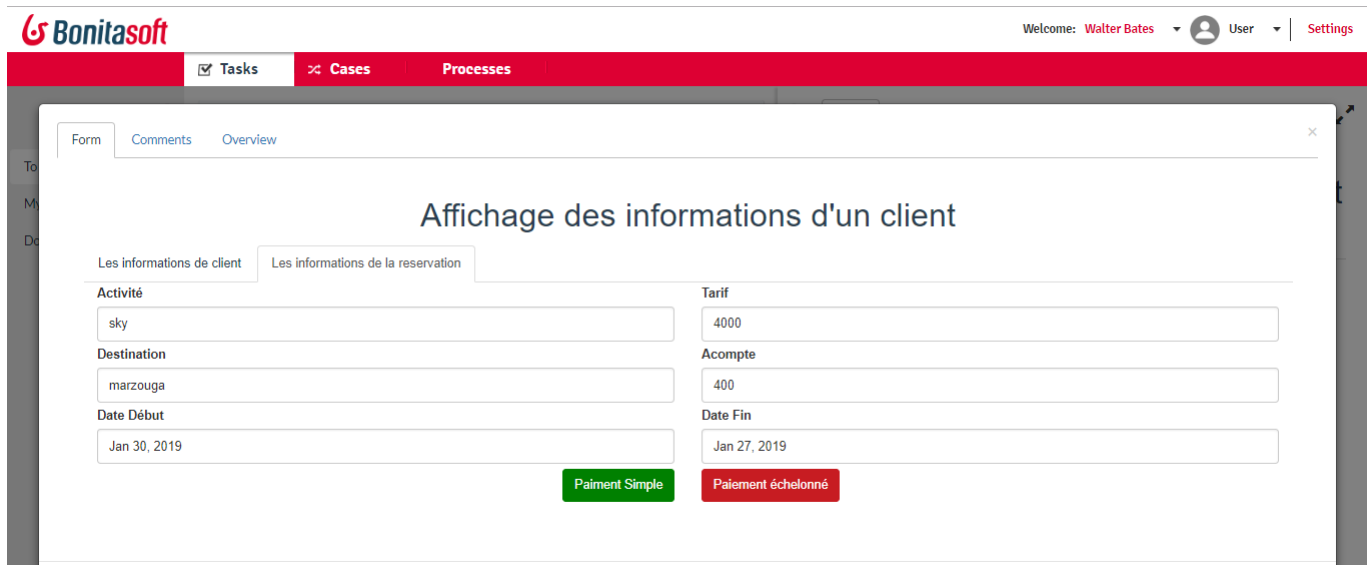
Informations Client	Informations Reservation
Destination* <input type="text" value="Marrakesh"/>	Activite* <input type="text" value="tourisme"/>
Acompte* <input type="text" value="4000"/>	Tarif* <input type="text" value="40000"/>
Date Du Debut de séjour * <input type="text" value="01/19/2019"/> Today	Date De la Fin du séjour * <input type="text" value="02/09/2019"/> Today
<input type="button" value="Valider"/>	

FIGURE 4.1.2 – saisir les informations de la réservation

Avant de choisir le type de paiement, on affiche les informations saisie par le vendeur lors de la première étape

The screenshot shows the Bonitasoft web application interface. At the top, there is a navigation bar with the Bonitasoft logo on the left and user information on the right: 'Welcome: Walter Bates', a user profile icon, 'User', and a 'Settings' link. Below the navigation bar is a red header with three tabs: 'Tasks' (selected), 'Cases', and 'Processes'. The main content area is a modal window titled 'Affichage des informations d'un client'. It has three sub-tabs: 'Form' (selected), 'Comments', and 'Overview'. The 'Form' tab contains two sections: 'Les informations de client' and 'Les informations de la reservation'. The 'Les informations de client' section has two input fields: 'Nom' with the value 'Boudouar' and 'Prenom' with the value 'Mohamed'. The 'Les informations de la reservation' section has two input fields: 'Email' with the value 'mo@gmail.com' and 'Téléphone' with the value '0600476989'. Below these input fields are two buttons: 'Paiement Simple' (green) and 'Paiement échelonné' (red).

FIGURE 4.1.3 – Affichage des informations de clien



Bonitasoft Welcome: **Walter Bates** User Settings

Tasks Cases Processes

Form Comments Overview

Affichage des informations d'un client

Les informations de client Les informations de la reservation

Activité
sky

Destination
marzouga

Date Début
Jan 30, 2019

Tarif
4000

Acompte
400

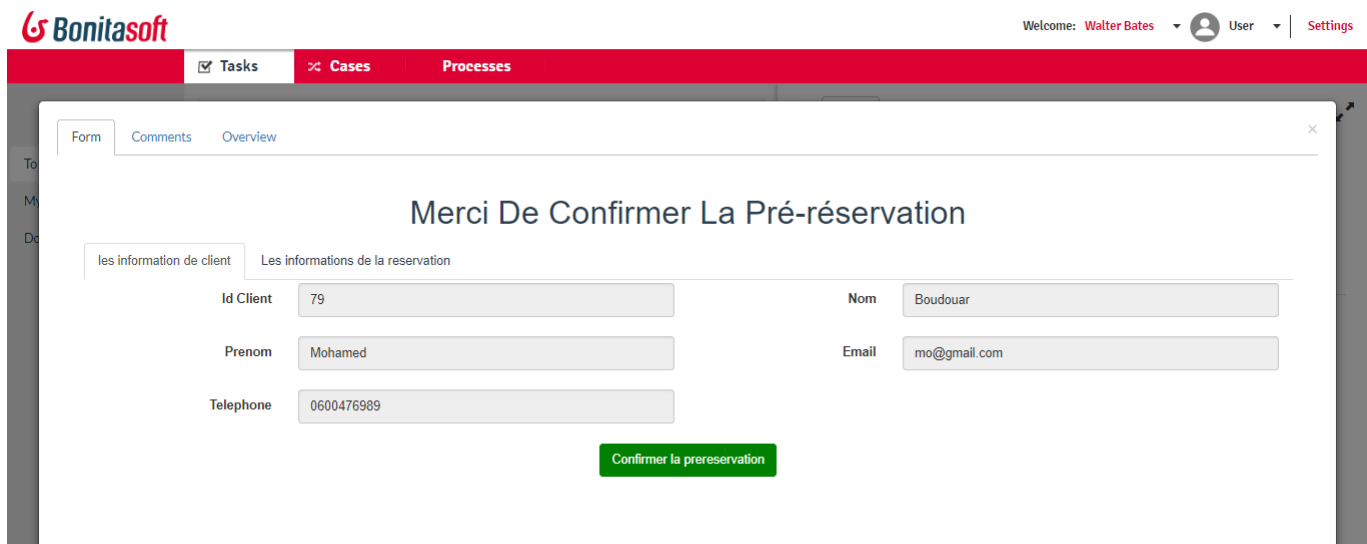
Date Fin
Jan 27, 2019

Paiement Simple Paiement échelonné

FIGURE 4.1.4 – Affichage des informations de la réservation

La deuxième étape est de choisir le type de paiement, entre le paiement simple ou bien le paiement échelonné.

Après avoir choisi le paiement échelonné, le vendeur confirme la pré-réservation



Bonitasoft Welcome: **Walter Bates** User Settings

Tasks Cases Processes

Form Comments Overview

Merci De Confirmer La Pré-réservation

les information de client Les informations de la reservation

Id Client
79

Prenom
Mohamed

Telephone
0600476989

Nom
Boudouar

Email
mo@gmail.com

Confirmer la prereservation

FIGURE 4.1.5 – Confirmer la pré-réservation

Dans cette étape on procède à la préparation de la demande de paiement échelonné. le vendeur demande au client les informations sur l'adresse, profession et le salaire afin de traiter ça demande de paiement échelonné.

Bonitasoft Welcome: **Walter Bates** User Settings

Tasks Cases Processes

Form Comments Overview

Préparation de la demande de paiement Echelonne

Id Client: 79 Num Reservation: 34

Veuillez fournir les renseignements suivants

Adresse*
saint louis MO

Profession*
Professor

Revenue*
50000

FIGURE 4.1.6 – Préparation de la demande

4.1.1 la demande de paiement échelonné acceptée

Le premier cas, la demande de paiement échelonné est positive c-a-d le client n'est pas un mauvais payeur ne figure pas sur la liste noire

Bonitasoft Welcome: **Walter Bates** User Settings

Tasks Cases Processes

Form Comments Overview

Votre demande de paiement echelonne a été approuvée

ID client: 79 Id reservation: 34

Profession: Professor Adresse: saint louis MO

Revenue: 50000

Procéder à la Réservation Définitive

FIGURE 4.1.7 – Préparation de la demande

DANS CE CAS IL NOUS RESTE QUE L'ÉTAPE DE LA RÉSERVATION DÉFINITIVE

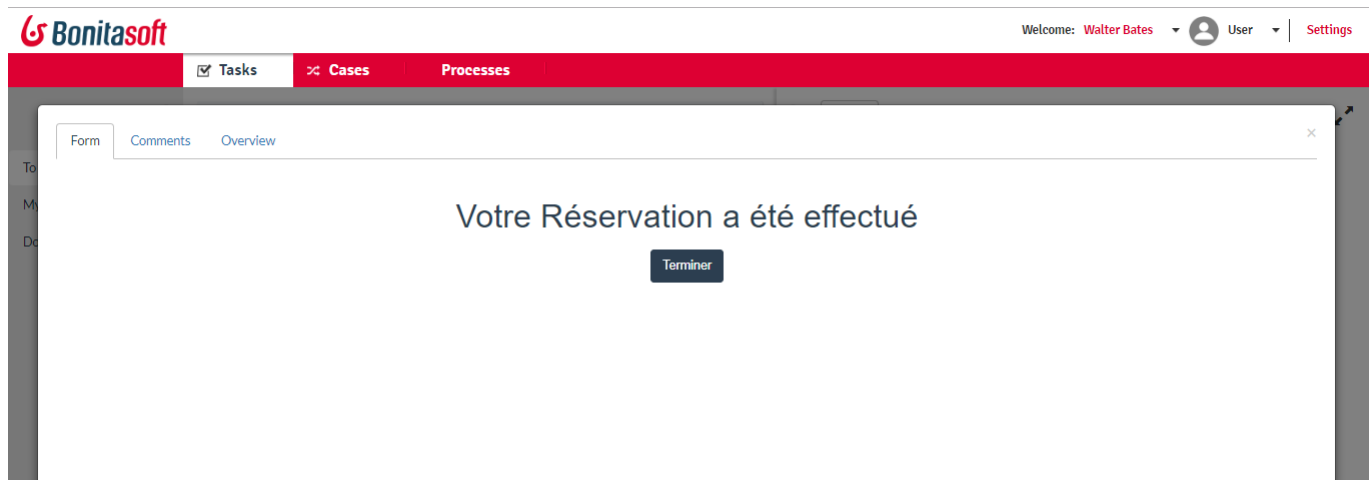



FIGURE 4.1.8 – Préparation de la demande

4.1.2 Le cas d'un mauvais payeur

Dans cette démonstration, on va choisir un mauvais payeur dans notre base de données. et le vendeur vas saisir les information de ce client pour une nouvelle réservation.

Formulaire du client



Informations Client	Informations Reservation
<p>Nom*</p> <input type="text" value="boudouar"/>	<p>Prenom*</p> <input type="text" value="mohamed"/>
<p>Email*</p> <input type="text" value="mboudouar@gmail.com"/>	<p>Telephone*</p> <input type="text" value="06004769898"/>
<p>Valider</p>	

FIGURE 4.1.9 – Préparation de la demande

On suite on vas procéder aux même étape le de première démonstration et notre système a détecté que le client est un mauvais payeur. dans ce cas le client a le choix entre payer la totalité de prix de voyage immédiatement ou-bien il demande le remboursement sur l'acompte.

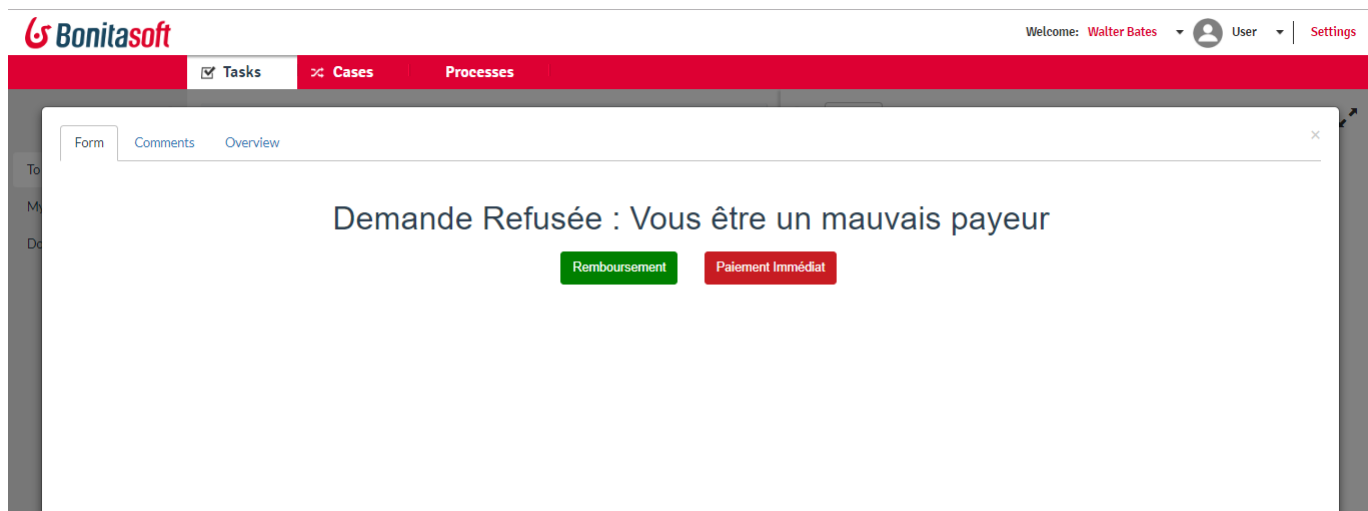


FIGURE 4.1.10 – Demande refusée

Le premier cas paiement immédiatement :

The screenshot shows the BonitaSoft web application interface for the 'Paieement Immédiat' (Immediate Payment) form. The form is titled 'Paieement Immédiat' and contains several input fields: 'ID Client' with the value 79, 'Num Reservation' with the value 35, 'Acompte' with the value 400, and 'Tarif' with the value 4000. Below these fields, the text 'il vous reste 3,600 à payer' (you still have 3,600 to pay) is displayed. There is also a 'Prise de la somme' (Sum of the sum) field. At the bottom of the form, there is a 'Confirmer' (Confirm) button. The interface includes a top navigation bar with a red header containing the BonitaSoft logo and the text 'Welcome: Walter Bates'. Below this, a secondary navigation bar has tabs for 'Tasks', 'Cases', and 'Processes'. The main content area is titled 'Paieement Immédiat'. The interface also shows a sidebar on the left with a list of items and a top right corner with a user profile icon and 'Settings' link.

FIGURE 4.1.11 – Paiement immédiat

Le deuxième cas demande de remboursement :

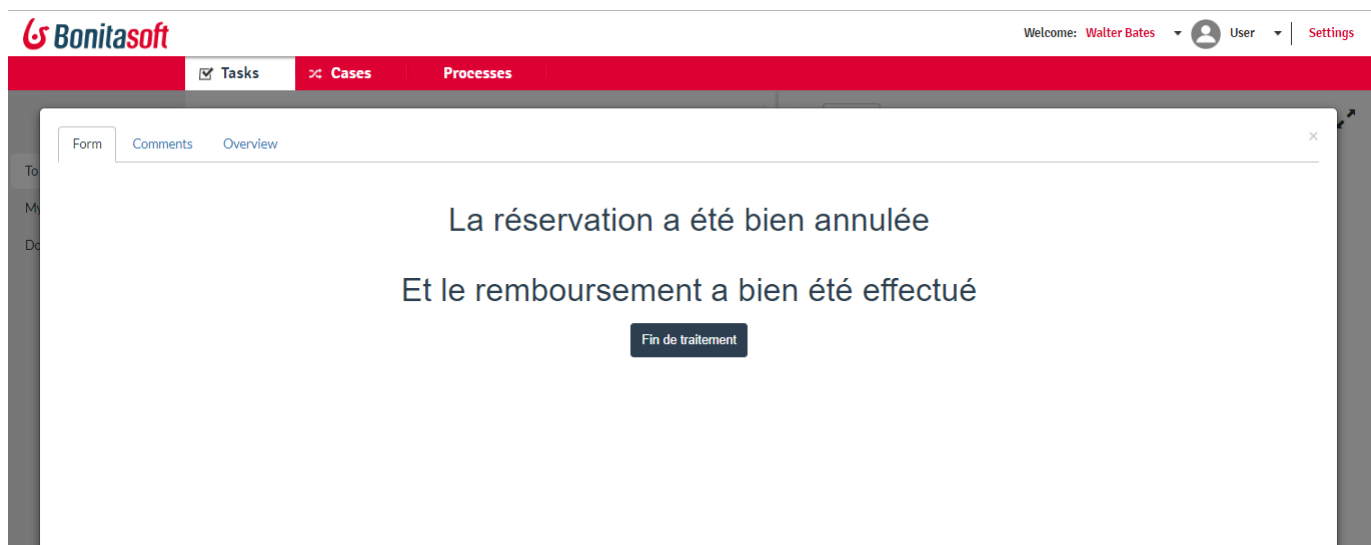


FIGURE 4.1.12 – Remboursement

4.2 Paiement des échéances

Pour régler le solde des échéances, le vendeur demande aux clients leur ID, et le numéro de la réservation.

The screenshot shows a web form titled 'Espace Facture et Paiement des echeances'. It contains two input fields. The first field is labeled 'Entrer le Numero de Client*' and the second field is labeled 'Entrer le Numero De La Réservation*'. Both fields are empty. Below the second field is a dark button labeled 'Submit'.

FIGURE 4.2.1 – Identification

L'étape suivante c'est l'affichage du bilan de paiement et la facture

The screenshot shows the Bonitasoft web application interface. At the top, there is a navigation bar with the Bonitasoft logo on the left and user information 'Welcome: Walter Bates' and 'User' on the right. Below the navigation bar, there is a red header bar with tabs for 'Tasks', 'Cases', and 'Processes'. The main content area displays a form titled 'Bilan de paiement--Facture--'. The form has two columns of input fields. The left column contains 'Num Client' (77), 'Acompte' (400), 'Montant' (4000), and 'Le Reste à payer' (2000). The right column contains 'Num Reservation' (33), 'Somme payée' (2000), and 'Numero Facture' (12). A 'Submit' button is located at the bottom center of the form.

Field	Value
Num Client	77
Acompte	400
Montant	4000
Le Reste à payer	2000
Num Reservation	33
Somme payée	2000
Numero Facture	12

FIGURE 4.2.2 – Bilan Des Paiements

Paieement des échéances

The screenshot shows the Bonitasoft web application interface. At the top, there is a navigation bar with the Bonitasoft logo on the left and user information 'Welcome: Walter Bates' and 'User' on the right. Below the navigation bar, there is a red header bar with tabs for 'Tasks', 'Cases', and 'Processes'. The main content area displays a form titled 'Espace de Paiement des echeances'. The form has three input fields: 'Num Client*' (77), 'Num Reservation*' (33), and 'Num Facture*' (12). Below these fields, there is a text message 'il vous reste 2,000 à payer'. At the bottom, there is a label 'Echeance/somme*' followed by an empty input field.

Field	Value
Num Client*	77
Num Reservation*	33
Num Facture*	12

il vous reste 2,000 à payer

Echeance/somme*

FIGURE 4.2.3 – Échéance de 1000 DHs

APRÈS AVOIR PAYER 1000 DHS DANS UNE ÉCHÉANCE.

The screenshot shows the Bonitasoft web interface. The top navigation bar is red with the Bonitasoft logo on the left and 'Welcome: Walter Bates', 'User', and 'Settings' on the right. Below the navigation bar, there are tabs for 'Tasks', 'Cases', and 'Processes'. The main content area is titled 'Espace de Paiement des echeances'. It contains a form with the following fields: 'Num Client*' (77), 'Num Reservation*' (33), 'Num Facture*' (12), and 'Echeance/somme*' (0). A message 'il vous reste 1,000 à payer' is displayed below the 'Num Facture*' field.

FIGURE 4.2.4 – Échéance de 1000dhs

LE PAIEMENT DE LA DERNIÈRE ÉCHÉANCE.

The screenshot shows the Bonitasoft web interface. The top navigation bar is red with the Bonitasoft logo on the left and 'Welcome: Walter Bates', 'User', and 'Settings' on the right. Below the navigation bar, there are tabs for 'Tasks', 'Cases', and 'Processes'. The main content area is titled 'Bilan de paiement--Facture--'. It contains a form with the following fields: 'Num Client' (77), 'Acompte' (400), 'Montant' (4000), 'Le Reste à payer' (0), 'Num Reservation' (33), 'Somme payée' (4000), and 'Numero Facture' (12). A 'Submit' button is located at the bottom right of the form. A blue notification box at the top of the form area states: 'Form submitted. The next task in the list is now selected.'

FIGURE 4.2.5 – Facture de Fin de Paiement

LA FIN DE PROCESSUS PAIEMENT DES ÉCHÉANCES.



FIGURE 4.2.6 – Règlement du solde, Fin de Paiement

Conclusion

Là où les architectures précédentes demandaient des ressources importantes, en terme de développement comme de déploiement, les Web Services héritent du pragmatisme des technologies du Net, ce qui facilite leur adoption. Mais les Web Services sont encore en cours de stabilisation des standards. Il sera notamment nécessaire de définir des normes de sécurité qui posent un gros problème. Cependant les Web Services gagnent en maturité ce qui peut promettre qu'ils seront adoptés massivement comme la réponse appropriée aux problématiques d'échanges de données et d'intégration d'applications. Il est donc prévu une augmentation d'utilisation des Web Services par les entreprises dans les années à venir.

Durant ce projet, la réalisation des différents défis de la mise en oeuvre d'une application se sont avérées très enrichissantes pour l'aspect technique aussi bien pour les disciplines des gestionnaires de projet.