Institut **S**upérieur
d'**I**nformatique, de
Modélisation et de
leurs **A**pplications

BP 10125
63173 Aubière Cedex

Sopra Groupe

Parc Technologique de la Pardieu
Immeuble Vision II
3 Allée Pierre de Fermat
63170 Aubiere

Rapport d'ingénieur

Stage de 3ème année

Filière : Génie Logiciel et Systèmes Informatiques

Stage à Sopra Group

Présenté par : Julien PINGUET
Sous la direction de : Claude MAZEL

Avril-Septembre 2013

Remerciements

Je tiens à remercier les personnes qui m'ont accueilli et accompagnés durant ces 6 mois de stage chez Sopra Group.

Je remercie tout d'abord Delphine JARIGE, chef de projet du projet *Sides*, ainsi que les autres membres de l'équipe, pour mon accueil en ce début de stage.

J'adresse aussi mes remerciements à Laurent LAFFERRÈRE, chef de projet du projet *LimaGest* sur lequel j'ai travaillé, ainsi que l'architecte Xavier CLEMENCE pour son aide apporté lors des difficultés rencontrées.

Enfin, je remercie l'équipe du projet de l'*Imprimerie Nationale* : Pierre ? ? le chef de projet, Ambroise ROQUETTE l'architecte, mais aussi les développeurs, pour l'aide apporté lors de mon affectation au projet.

Table des figures

Résumé

Mon stage de fin d'études à l'ISIMA avait pour but l'intégration d'une société de services en ingénierie informatique, affecté à un projet de refonte d'une application de recrutement de saisonniers.

Tout d'abord, j'ai étudié les besoins du client, ainsi que l'ancienne solution, permettant de modéliser le modèle de données et la nouvelle application.

Après le choix des technologies, j'ai développé une première version fonctionnelle de l'application. Celle-ci a permis au client d'énumérer les problèmes et proposer des améliorations ergonomiques et fonctionnelles.

Une fois l'application opérationnelle et son déploiement effectué, il ne restait plus qu'à importer les données de l'ancienne base de données pour permettre une transition efficace.

Mots clé : Sopra Group, Microsoft, VB.NET, Silverlight, SQL Server

Abstract

TODO

Keywords : Sopra Group, Microsoft, VB.NET, Silverlight, SQL Server

Table des matières

Glossaire

TODO : commandes LaTeX

Bibliothèque : TODO

Framework : ensemble de composants logiciels permettant le développement le développement rapide d'applications.

Introduction

Dans de nombreux contextes le gain de temps est le maître mot. Le domaine professionnel ne fait pas exception à la règle, imposant une productivité de plus en plus élevée. Ainsi l'utilisation d'outils de plus en plus performants permet de minimiser le temps passé à effectuer une tache.

Limagrain utilise un outil un outil de recrutement, développé il y a plus de dix ans par un membre de service ne possédant de compétence en développement. Dû au nombre de recrutements de plus en plus élevé, et l'augmentation de la charge de travail, Limagrain désire faire évoluer son outil pour faciliter le processus actuel.

Le passage des années, ainsi que l'évolution des besoins, impose une mise à niveau permanente des outils informatiques. Mais lorsque celui-ci utilise des technologies trop anciennes, et ne possède pas une architecture évolutive, il est rarement possible d'intervenir. La meilleure solution est donc une ré-implémentation complète de l'outil, amenant alors à une solution structurée qui pourra évoluer facilement aux cours des années.

L'objet de cette étude sera d'étudier les besoins du client, puis d'implémenter la nouvelle solution. Celle-ci devra remplir les mêmes fonctions que la solution actuelle, en permettant l'ajout de nouvelles.

L'étude de la solution actuelle et l'étude des besoins de l'utilisateur permettra de modéliser la nouvelle solution. L'implémentation d'une première version de l'application permettra de soulever les problèmes existants et d'émettre les points d'amélioration. Enfin la mise en production et l'importation des anciennes données a permis au client d'utiliser l'application.

Tout d'abord, j'analyserai précisément le sujet, en présentation l'entreprise, ainsi que la solution actuelle et celle envisagée. Dans un deuxième temps, je présenterai l'environnement de travail et la structure du projet. Pour terminer, je présenterai la solution obtenue et les différents points d'amélioration possibles.

I Introduction de l'étude

I.1 Présentation de l'entreprise

I.2 Le métier d'ingénieur

I.3 Étude du problème

I.3.1 Besoin de l'utilisateur

Durant l'été, saison des récoltes des céréales, Limagrain emploie de nombreux travailleurs. Il s'agit principalement de contrats saisonniers, à durée déterminée, qui seront

Les candidats effectuent souvent plusieurs contrat dans l'entreprise. Il est donc nécessaire de mémoriser leurs informations pour pouvoir les réutiliser les années suivant et de les mettre à jour en cas de besoin.

De plus, la création de contrat s'effectuent rapidement, du fait que les agriculteurs ne peuvent prévoir les demandes de ressource à l'avance. Le service des ressources humaines doit ainsi saisir de nombreux informations sur les candidats et éditer les contrats très rapidement.

I.3.2 Solution actuelle

Le service des ressources humaines utilisait un document Access, de Microsoft Office. Ce document comporte une base de donnée ainsi qu'une interface de saisie.

Cette solution comporte de nombreux inconvénients.

Tout d'abord, la saisie des données est fastidieuse car l'ergonomie est très sommaire : les champs sont disposés de manière désordonnée sur la page. Par exemple lorsque l'utilisateur doit saisir un établissement de travail, il doit saisir son code exacte, sans commettre d'erreur.

Les fonctionnalités sont limitées, et se limitent principalement à la saisie.

I.3.3 Solution envisagée

En étudiant la solution actuelle, les chefs de projet ont réfléchis à la meilleure solution qui corrigerait ses inconvénients, et apporterait de nouveaux avantages.

Il es tout débord nécessaire de changer de base de données pour se tourner vers un système plus performant. Comme le groupe Limagain est tourné vers les technologies Microsoft, il a été décidé d'utiliser *Microsoft SQL Server*.

Pour améliorer l'ergonomie de l'application, la nouvelle solution sera un *client léger*. En effet, nous ne seront pas confrontés aux problèmes de performances qu'offre cette solution, et en cas de défaillance du serveur l'application sera inutilisable dans tous les cas du fait que la base de données y sera centralisée. Ainsi, un simple site web offre les avantages d'un déploiement unique, d'une maintenance simple.

——————————— Access : sécurité, maintenance, évolutivité au moins 1000 candidatures en 1 mois

II Conception de la solution

II.1 L'équipe du projet

Pour réaliser ce projet, j'ai été intégré dans une petite équipe de trois personnes.

Un chef de projet qui est le relai entre Sopra Group et Limagrain affin d'établir la compréhension du besoin fonctionnel et technique. Sa charge est d'une demi-journée par semaine. De plus, il est chargé de la rédaction des spécifications de l'application. Pour des raisons de délais trop courts et charges limitées, ces spécifications ont été réalisées en parallèle du développement, malgré l'éloignement des bonnes pratiques.

Un architecte a pour objectif de suivre le développeur, conseillant les outils, technologies et méthodes utilisées. Sa charge sur le projet est aussi d'une demi-journée par semaine.

Et enfin moi même, en tant que développeur, en charge du développement de l'application, à plein temps.

II.2 La relation client

II.2.1 Assistance technique

Le projet s'est réalisé en *assistance technique*. Il n'y a aucune date de livré imposée, par contre la charge (nombre de jours d'interventions) totale est fixe.

J'ai travaillé à raison de trois jours chez le client Limagrain et deux jour au siège de Sopra Group, profitant ainsi de la proximité du client pour éclaircir les points ambigus.

II.2.2 Définition du besoin

La relation directe avec le client m'a permis de définir le besoin du client au cours des différentes étapes du projet.

La première étape a consisté à établir le modèle de données. Pour cela il a fallu étudier la solution existante pour déterminer les informations à mémoriser, leur type, les contraintes, ainsi que les liens entre elles. Cette partie est importante car la structure de l'application est fortement liée a ce modèle.

Deuxièmement, j'ai réalisé une première version de l'interface graphique de l'application. L'objectif est de proposer une première maquette au client qui pourra effectuer des critiques. L'ergonomie de la solution s'est ainsi optimisé au fur et à mesure des démonstrations.

Enfin, une fois l'application fonctionnelle, il était nécessaire d'importer les données existantes dans la nouvelle de données. Comme ces deux bases de données sont différentes, il était nécessaire d'être en contact avec le client pour faire correspondre les schémas.

II.3 Technologies et architecture

Dans cette section je vous présenterai les différentes technologies utilisées dans le projet ainsi que leurs interactions. La figure **??** schématise les différents composants et communications.

Le groupe Limagrain travaille dans un environnement Micosoft et utilise ainsi leurs technologies et logiciels, comme par exemple : Windows pour le système d'exploitation, Internet Explorer comme navigateur internet, Outlook comme messagerie, ...

Pour développer cette solution nous nous sommes donc tourné vers les technologies Microsoft, facilitant ainsi la compatibilité et l'intégration des composants.

L'architecture d'une application est importante, car cela définie sa maintenabilité et son l'évolutivité. De plus cela permet la séparation des problèmes diminuant ainsi la complexité.

Notre application est divisée en 3 couches, appelé *3-tiers*, qui est le modèle multi-tiers le plus utilisé. Cela permet de séparer l'accès aux données de la base, la partie métier effectuant les traitements, et l'interface de l'utilisateur, comme le représente la figure **??**.

II.3.1 Socle

La solution développée se base sur un socle existant, développé dans le cadre d'un autre projet. Ceci nous a permis un gain de temps important car une partie importante du projet n'a du être développé à nouveau. En contre partie, les différentes technologies utilisées nous ont été imposées.

II.3.1.1 Gestion des utilisateurs

Le socle permet une gestion des utilisateurs de l'application. Il est possible de les créer, modifier ou supprimer, pour restreindre l'accès aux personnes autorisées. De plus, il est possible de connecter l'application à un LDAP.

Le *LDAP*, pour Lightweight Directory Access Protocol, est un protocole standard de gestion d'utilisateurs. Son objectif est de centraliser les informations des utilisateurs (nom, identifiant, mot de passe, ...) dans un annuaire.

Le principal avantage de cette solution est la mise en commun des comptes utilisateurs, ce qui permet l'utilisateur d'un seul et même compte pour tous les services connectés : Windows, la boite mail Outlook, cette application, ...

II.3.1.2 Gestion des droits

Il est aussi possible de gérer des droits dans l'application grâce à ce socle. L'administrateur peut ainsi contrôler les accès et les actions des utilisateurs, protégeant ainsi les informations confidentielles.

Chaque écran de l'application peut avoir leur accès ou certaines actions restreintes en fonction d'un *droit*. Ceux-ci peuvent prendre trois valeur :
- "aucun accès", par défaut, qui interdit tout accès à l'écran ;
- "lecture" n'autorise que l'accès à l'écran, avec la possibilité d'effectuer des recherches ou d'afficher les détails ;
- "lecture et écriture" autorise toutes les actions possibles.

Un *profil* est un ensemble de droits, qui permet de définir un périmètre d'action. Par exemple un profil "administrateur" aura tous les droits dans l'application, un profil "manager" n'aura que des droits de lecture, ou encore un profil "ressource humaine" possèdera les droits liés aux candidats et contrats, ...

Chaque utilisateur possède un ou plusieurs profils, ce qui permet de définir l'ensemble de ses droits. L'utilisation de profils plutôt que de directement de droit permet un gain de temps, car la personne administrant les utilisateurs n'aura pas à affecter les nombreux droits aux nouveaux utilisateurs, et la modification d'un profil permet d'impacter l'ensemble des utilisateurs associés.

La figure **??** représente le schéma UML des utilisateurs, profils et droits dans l'application.

II.3.1.3 Programmation modulaire

Un module est un projet indépendant effectuant une tâche particulière. Le socle offre la possibilité de programmer de façon modulaire, où chacun des modules ne possèdent (quasiment) aucunes interactions entre elles.

Ainsi plusieurs modules ont été développés, effectuant des tâches distinctes.

II.3.2 Langage de programmation

II.3.2.1 Microsoft .NET Framework

.NET Framework est une plateforme application proposée par Microsoft. Cette technologie est comparable et directement concurrente de Java d'Oracle.

Ce framework est constitués de nombreux composants, disposés en couches au fur et à mesure de ses versions, schématisés sur la figure **??** :

1. Le moteur d'exécution appelé Common Language Runtime (CLR) permet de compiler le code course en un langage intermédiaire appelé Microsoft Intermediate Language (MSIL). Ce code est ensuite compilé à la volée lors de la première exécution grâce au compilateur "Just In Time" (JIT) ;
2. Une bibliothèque de classes, offrant des fonctionnalités de base pour les différentes applications ;
3. Plusieurs couches supplémentaires proposant des outils de développement d'interfaces graphique (WinForms), d'accès aux données (ADO.NET (Entity Framework)), ...

II.3.2.2 VB.NET

Visual Basic .NET est un langage de programmation développé par Microsoft. Il s'agit d'une évolution majeure de Visual Basic 6, introduisant notamment l'aspect orienté objet. De plus, le code est compilé dans le langage intermédiaire que les différents langages fonctionnant sur la machine virtuelle .Net. Ce langage est très proche du C#, à la syntaxe près.

Le projet a été développé dans ce langage, aussi bien la couche métier que dans la couche présentation.

II.3.2.3 Apache log4net

Les *log* sont les informations mémorisées lors du fonctionnement d'un programme informatique. Ils permettent de garder des traces d'exécution, notamment lors du démarrage, de l'arrêt, ou d'erreurs. Les log peuvent faciliter le débogage et la recherche d'anomalies. Un ou plusieurs fichiers texte sont générés donnant accès à ces informations.

Le framework *log4net* (TODO : footer) est un port sous Microsoft .NET du projet libre et open-source "Apache log4j". Il permet de gérer simplement les log dans l'application.

II.3.3 Base de données

Une *base de données* permet de stocker un grand nombre d'informations ayant des natures différentes. Les données de même nature sont stockées dans des tables, où un champs possède un type et représente une valeur.

II.3.3.1 PowerAMC

PowerAMC, version francophone de PowerDesigner, est un logiciel de modélisation de base de données produit par la société Sybase. Il permet d'établir facilement un modèle de données à partir de son interface graphique, visible sur la figure **??**. De plus, les scripts générés sont compatibles avec les différents types et versions de bases de données, ce qui permet se s'abstraire de certaines contraintes de compatibilité.

Ce logiciel est utilisé par Sopra Group sur les différents projets. Je l'ai donc utilisé sur ce projet. Il m'a permis de concevoir, mais aussi effectuer les différentes modifications très simplement.

II.3.3.2 SQL Server

Il existe de nombreux système de gestion de base de données : Oracle, MySQL, PostgreSQL, ...Nous avons utilisé *Microsoft SQL Server*, par demande du client car il s'agit d'un standard pour Limagrain.

II.3.3.3 Mapping de la base de données

Définition

Le "modèle relationnel" est utilisé dans les systèmes de gestion de base de données (SGBD) pour rassembler un ensemble d'informations. Les données (clés) sont dupliquées entre les tables et l'accès aux relations s'effectue ensuite grâce à des jointures entre les tables.

Le "modèle objet", quant à lui, est utilisé dans la programmation orientée objet. Les données sont modélisées sous la formes d'objets, entités complexes ayant des comportements et des relations entre elles.

Le *mapping objet-relationnel* consiste à interfacer le modèle relationnel d'une base de données avec le modèle orienté objet d'un programme informatique. Généralement, une classe modélisera une table, et attribut d'objet modélisera un champ d'une table, avec un type similaire (par exemple `String` pour `varchar`).

Entity Framework

Cette opération peut être faite à l'aide d'un framework (TODO : glossaire), permettent de s'abstraire de la base de données, automatisant et réduisant ainsi la duplication de code. L'objectif est de faciliter le développement, augmenter la maintenabilité du programme, ou encore s'abstenir du type de base de données. Ainsi, la réaction, la recherche, la mise à jour, ou la suppression (opérations CRUD) de données se font de manière simplifiées voir transparentes pour l'utilisateur.

Microsoft propose plusieurs framework, et c'est *Entity Framework* qui a été utilisé. Il est intégré à Visual Studio, ce qui permet une génération et un paramétrage facile d'utilisation. De plus, il permet l'interaction avec LINQ (Language-Integrated Query), extension du langage permettant de faire des requêtes sur des ensembles de données en s'abstrayant du type.

II.3.4 Les services

La couche de *service* est la partie métier de l'application. Elle permet de séparer le code source lié à l'affichage destiné à l'utilisateur, du code métier effectuant des traitements dans l'application.

II.3.4.1 Client-Serveur

Nécessité

Il est nécessaire de faire communiquer la couche cliente de l'application avec la couche métier en raison de l'utilisation de Silverlight, comme je l'expliquerai dans la section II.3.5.1 ci-après. Ces deux couches ne peuvent communiquer directement entre elles car l'une est exécutée sur le serveur et l'autre sur l'ordinateur de l'utilisateur. La solution consiste à utiliser un service web.

Web service

Un *service web* (ou *web service*) est un programme informatique permettant la communication et l'échange d'informations entre des systèmes hétérogènes et distribués (local, réseau, internet, ...), exposant des fonctionnalités.

L'échange d'informations entre le client et le serveur se fait par sérialisation, consistant à coder les informations contenues en mémoire. Cela peut se faire sous le format texte (XML, JSON, ...) ou même au format binaire. L'objectif est d'échanger des données génériques et abstraites qui pourront être représentées dans n'importe quel langage.

L'exemple qui suit montre un exemple de sérialisation d'un même objet sous le format XML :

```
<menu id="file" value="File">
    <popup>
        <menuitem value="New" onclick="CreateNewDoc()" />
        <menuitem value="Open" onclick="OpenDoc()" />
        <menuitem value="Close" onclick="CloseDoc()" />
    </popup>
</menu>
```

et JSON :

```
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

Windows Communication Foundation

Windows Communication Foundation, couramment appelé sous ses initiales WCF, est la couche de communication de .NET Framework, apparue dans la version 3.0. Cette technologie respecte les normes standards des services web, ce qui lui permet d'appeler ou d'être appelé par des technologies différentes (Java, Python, ...).

Ce framework a été utilisé pour réaliser le service web de l'application, exposant ainsi différentes fonctions utiles à la couche d'affichage.

II.3.4.2 WCF RIA Services

Il est souvent nécessaire de posséder une logique applicative à la fois du coté serveur et du coté client. C'est le cas par exemple lorsque l'on souhaite vérifier la validité des données avant de les insérer en base de données :

1. Soit on effectue la vérification coté serveur, impliquant l'échange inutile d'informations lorsque celles-ci sont fausses ;
2. Soit la vérification est faite coté client, ce qui empêcherait le bon contrôle des flux d'entrée avant le traitement, ce qui pourrait amener à des erreurs ;
3. Soit on effectue la même vérification à la fois du coté client et du coté serveur, mais cela impose de dupliquer le même code dans les deux couches.

Pour éviter ce problème, Microsoft propose le framework *WCF RIA Services*. Cet outil duplique du code d'un projet pour le copier dans un autre, de façon automatiquement lors de la compilation. L'opération est schématisé sur la figure **??**. Ceci permet d'avoir un code métier constamment à jour sans avoir à effectuer les modifications dans les différentes couches.

II.3.4.3 Reporting services

Microsoft fourni un outil de génération de rapports appelés *Reporting services*. Cet un outil est intégré à la suite SQL Server. Les fichiers source utilisés par le service se présentent sous la forme d'un fichier XML. Visual Studio propose un éditeur graphique simplifiant son édition, la mise en forme ainsi que la création de requêtes.

Le fonctionnement du service est représenté par la figure **??** et se déroule en plusieurs étapes :

1. On appelle le service à l'aide de son adresse IP ou son nom de domaine. Il est possible de passer des paramètres dans l'URL en utilisant la méthode GET. Par exemple, si le service est accessible à l'URL `http://nomserveur/reports/`, l'appel de `http://nomserveur/reports/ID=5&name=toto` permet de spécifier les paramètres `ID` et `name` avec les valeurs respectives `5` et `toto`.
2. Le serveur effectue les requêtes nécessaires dans la base de données afin de récupérer les données dynamiques du rapport.
3. Le rapport est généré puis retourné au client au format HTML, PDF ou autre.

Ce type de service a été utilisé dans le projet. Il est ainsi facile de générer les divers documents à imprimer avec les informations correspondantes aux contrats et candidats. Le serveur de rapports est installé sur la même machine que celle de la base de données, pour ne pas avoir à installer inutilement SQL Server, mais aussi parce que tous les serveurs sont installés sur la même machine.

II.3.5 Interface utilisateur

II.3.5.1 Silverlight

Microsoft *Silverlight* est un plugin pour navigateur web. Il permet le développement d'applications riches et de pousser plus loin l'expérience utilisateur du web 2.0, au même titre qu'Adobe Flash dont il se veut une alternative. Initialement prévu pour des applications web dans un navigateur, les programmes peuvent être téléchargées pour être utilisés directement sur l'ordinateur ("out of browser"), et permettent aussi le développement d'applications pour Windows Phone 7.

Cette technologie nécessite l'installation du plugin, qui est un sous-ensemble de Microsoft .NET Framework. Les applications sont ainsi cross-browser (Internet Explorer, Firefox, Chrome, ...) et cross-platform (Windows, mais aussi OS-X et Linux via le projet open-source Moonlight). De plus les applications

fonctionnent dans une "sandbox" ("bac a sable") ce qui permet de garantir une sécurité accrue pour l'utilisateur.

Une des spécifications de Silverlight est l'impossibilité d'appeler des services web de manière synchrone, c'est à dire que la fonction appelante attend la réponse. En effet, lors d'un appel à un service le programme continue son exécution et un événement est émis lors de la réception de la réponse. Cette solution permet de ne pas bloquer l'interface de l'utilisateur qui pourrait penser à un plantage de l'application. Mais l'inconvénient est l'augmentation de la complexité de programmation car il est nécessaire de contrôler plusieurs fils d'exécution en parallèle.

II.3.5.2 ASP.NET

Un projet "Web Silverlight" nécessite la création d'un projet web, en plus du projet purement Silverlight. En effet, ce projet web permet d'encapsuler le Silverlight pour le transmettre au navigateur du client. Il est réalisé en *ASP.NET* qui est une technologie Microsoft de programmation web, permettant de créer des sites dynamiques.

Il est nécessaire d'utiliser un serveur web pour déployer le projet ASP.NET. Nous avons encore utilisé une technologie Microsoft : *Internet Information Services*, plus couramment appelé *IIS*.

II.3.5.3 SignalR

Le temps réel dans une application web permet au client de recevoir des données au moment où elles sont publiées. Un exemple courant est une discussion instantanée : le serveur doit informer le client lors de la réception d'un message, sans que celui-ci n'ait à rafraichir la page continuellement.

Avant l'apparition du HTML5, il était nécessaire d'utiliser des techniques JavaScript en envoyant continuellement des requêtes du client vers le serveur. Le HTML5 a apporté cette fonctionnalité grâce aux WebSocket, canal de communication bidirectionnel entre un navigateur web et un serveur web.

Microsoft fourni une bibliothèque intégrée à ASP.NET appelée *SignalR* (TODO : footer) permettant l'ajout de fonctionnalités de temps réel dans une application web. Elle est basée sur les WebSocket et du JavaScript, facilitant ainsi son utilisation.

TODO : barre de chargement ?

---

3 modules : candidats, contrat, paramétrage partie 3)

1 module = 1 truc indépendant

III Résultats - Discussions

III.1 Interface graphique utilisateur

J'ai été libre de programmer la première version de l'interface graphique, comportant les fonctionnalités demandées par le client. Ce dernier a participé à sa réalisation en effectuant diverses critiques et réagencements des éléments afin d'optimiser l'ergonomie de l'application.

Je présenterai dans cette section la structure de l'affichage et les différents types d'écran.

III.1.1 Structure

III.1.1.1 Barre supérieure

La partie supérieure de l'écran possèdes deux fonctions. La première est l'affichage des détails de l'application : logo, nom de l'application, version. La seconde est liée au compte de l'utilisateur : lien permettant de changer d'utilisateur, de déconnexion, et permettant de quitter l'application.

III.1.1.2 Menu latéral

Une barre latéral située sur la partie gauche propose un *menu*. Chaque ligne de ce menu correspond à un module de l'application. La figure **??** est une capture d'écran du menu, montrant les différents menus, et donc modules, de l'application.

De plus, un sous-menu de second niveau apparait lorsque l'on sélectionne un menu. Il s'agit de chacun des écrans principaux du module correspondant. Par exemple, le menu "Accès" possède quatre sous-menus correspondants à quatre fonctionnalités, comme le montre la figure **??**.

III.1.2 Les types d'écran

Les écrans peuvent peuvent être de plusieurs types.

Un écran d'affichage de données comporte un tableau permettant d'afficher une liste de valeurs. Il y a une zone de recherche permettant de filtrer les données. De plus, des boutons situés en bas de l'écran permettent d'effectuer des actions sur le ou les élément(s) sélectionné(s) : affichage des détails, modification, suppression, ...

L'affichage, la modification ou la création d'un élément est une fenêtre s'affichant par dessus l'écran actuel sous la forme d'un pop-up. Lorsqu'il s'agit d'une édition, les éléments sont des "zones de texte" ou des "listes déroulantes", sinon ce sont simplement des "labels".

Des écrans de "paramétrage" permettent la gestion des tables "statiques". Ce sont les valeurs des différentes listes déroulantes que l'on peut trouver dans les écrans de saisie. Il est ainsi possible d'administrer ces listes en supprimant certaines entrées ou en créant de nouvelles.

III.2 Base de données

Le modèle de données actuel satisfait bien le besoin du client. Il modélise correctement les différentes données à enregistrer, avec les contraintes et dépendances imposées.

Cependant il reste quelques améliorations possibles. En effet, le client a tendance à ajouter, modifier ou supprimer certaines spécifications au court du projet, imposant des changes plus ou moins importants dans le modèle. Ces changements impactent aussi les différentes couches de l'application, du mapping à l'interface graphique, ce qui peut faire perdre un temps important en fin de projet. Ainsi les modifications "non bloquantes" n'ont pas été apportées et le seront lors du lot 2.

TODO : DAPE, DUE

Conclusion

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Bibliographie

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Annexe

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.