



---

# TP .Net

---

Bug tracking like

---

Lemettre Arnaud

Version 1.0

7 Pages

02/04/2009

EPITA-MTI2014-NET-TP-bug-track

---



## Propriétés du document

<b>Auteur</b>	<b>Lemettre Arnaud</b>
<b>Version</b>	1.0
<b>Nombre de pages</b>	7
<b>Références</b>	EPITA-MTI2014-NET-TP-bug-track

## Historique du document

<b>Date de vision</b>	<b>Version</b>	<b>Auteur</b>	<b>Changements</b>
27/02/2009	0.1	Lemettre	création

## Site de référence

<b>description</b>	<b>url</b>
Site MTI	
Blog MTI	

## Sommaire

Introduction .....	4
Contexte.....	5
Partie 6.....	6
Travail a faire .....	6
Modalité de rendu .....	6

## Introduction

Le but de cette série de TP sera la réalisation d'un système de bug tracking. Ce tp se décompose en plusieurs parties. Toutes les semaines vous aurez une partie à rendre. Cette partie sera évaluée. Toute fois si pour une raison quelconque vous n'avez pas réussi à faire convenablement cette partie une correction sera postée une journée avant le début du prochain TP. Vous aurez à charge de l'intégrer dans votre code si celui ne produit pas le fonctionnement attendu.

Ce travail est à faire individuellement, tout code similaire sur deux personnes sera considéré comme un travail non rendu et non négociable.

Durant cette série de TP les notions suivantes seront abordées :

- modélisation BDD
- création de script SQL (Base, tables, données)
- création procédures stockées
- Entities Framework
- Linq To SQL
- Manipulation Open XML
- Génération de Flux XML
- Dynamic Data Entities
- Membership provider / Role provider
- utilisation des pages aspx et composants
- WCF (Webservice et sécurisation)

Chaque partie à rendre ne demande pas plus de 3h de travail par semaine, Bonne chance ;)

## Contexte

Un système de suivi de problèmes (de l'anglais bug tracking system ou système de suivi de bogues) est une application informatique qui permet d'aider développeurs et utilisateurs à améliorer la qualité d'un logiciel. Les utilisateurs soumettent les bogues (défauts de fonctionnement) rencontrés dans le logiciel. Les développeurs sont alors toujours au fait des problèmes rencontrés.

L'ensemble de la solution devra utiliser SQL Server 2012 pour la base de données, et le Framework 4.5 de .Net.

Le rendu de chaque lot, correspond au rendu de chaque partie.

## Partie 6

Nous allons créer un système de login, cependant nous avons déjà les informations de l'utilisateur présent en base de données.

Nous modifierons donc les membership pour pouvoir s'intégrer directement dans notre base de données.

### Travail à faire

Sur la page Default.aspx rajouter un système de login qui permettra de rentrer vos informations et de vous identifier sur le site. L'ensemble du site ne devra être accessible que si vous êtes présent dans la base de données. Il vous faut donc modifier le web.config pour que le système puisse prendre en compte ces changements.

Avant de modifier le fichier web.config il vous faut rajouter une classe dans le businessManagement pour implémenter un custom Membership provider. Cette classe devra se nommer CustomMemberShip.cs et permettre uniquement les utilisateurs contenus dans la table T\_User.



Figure 1 - Architecture de la solution

### Modalité de rendu

Les fichiers seront à rendre dans une tarball ayant pour nom :

login\_l.zip

Cette tarball devra comprendre :

- un script de création de votre base ainsi que des tables.  
Nom : CreateBugTrack.sql
- Un script pour remplir votre base avec des données  
Nom : DataBugTrack.sql
- Un dossier contenant la solution Visual Studio qui devra compiler.

Nom : login\_BugTrack

Le tout à envoyer sur l'adresse [mti.rendu.dotnet@gmail.com](mailto:mti.rendu.dotnet@gmail.com) avec les balises suivantes :

[MTI2014][NET][login\_l][BugTrack] partie 6