**TP Airport Management**

**Partie 8 Part2: ASP.NET MVC**

**V Importation d’une image**

1. On se propose d’utiliser l’image du pilote comme information supplémentaire pour identifier le vol.
   1. Dans la classe **Flight**, ajouter la propriété **Pilot** de type string.

public string Pilot { get; set; }

* 1. Mettre à jour la BDD.
  2. Adapter la vue **Create** pour l’upload de l’image.

<**form** **asp-action**="Create" enctype="multipart/form-data">

<div class="form-group">

<**label** **asp-for**="Pilot" class="control-label"></**label**>

<input type="file" name="PilotImage" class="form-control" />

<**span** **asp-validation-for**="Pilot" class="text-danger"></**span**>

</div>

* 1. Dans le projet **AM.UI.WEB**, sous le dossier **wwwroot** ajouter un nouveau dossier nommé **uploads**.
  2. Modifier l’action **Create POST** afin de télécharger l’image du pilote et l’insérer sous le répertoire **uploads** en plus de l’ajout du vol. Utiliser le code suivant :

Ajouter le paramètre correspondant à l’image parmis les paramètres de la méthode Create(Post)

public ActionResult Create(Flight collection, IFormFile PilotImage)

Dans le bloc try, sauvegarder l’image sous uploads et récupérer le nom de l’image dans la propriété Pilot du Flight avant de le sauvegarder

//sauvegarder l'image sous uploads

if (PilotImage != null)

{

var path = Path.Combine(Directory.GetCurrentDirectory(),

"wwwroot", "uploads", PilotImage.FileName);

Stream stream = new FileStream(path, FileMode.Create);

PilotImage.CopyTo(stream);

collection.Pilot = PilotImage.FileName;

}

1. Adapter la vue **Index** pour afficher l’image du pilote.

<td>

<**img** src="~/uploads/@item.Pilot" width="50" height="50">

</td>

1. Tester l’ajout d’un vol et le téléchargement des images.
2. Créer de la vue **Edit** qui permet de modifier les informations d’un vol.

**VI Update/Delete/Details**

Passer l’id du Flight dans les liens Edit, Delete et Détails de la vue Index

<**a** **asp-action**="Edit" **asp-route-id**="@item.FlightId">Edit</**a**> |

<**a** **asp-action**="Details" **asp-route-id**="@item.FlightId">Details</**a**> |

<**a** **asp-action**="Delete" **asp-route-id**="@item.FlightId">Delete</**a**>

1. Créer la vue **Delete** qui permet de supprimer un vol.

Generer la vue Delete

Passer l’objet à supprimer dans la méthode View de Delete(Get) afin qu’il soit affiché dans la vue.

// GET: FlightController/Delete/5

public ActionResult Delete(int id)

{

return View(sf.GetById(id));

}

Supprimer l’objet en question dans la méthode Delete(Post)

// POST: FlightController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, Flight collection)

{

try

{

sf.Delete(sf.GetById(id));

sf.Commit();

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

1. Créer la vue **Edit** qui permet de mettre à jour un vol.

Generer la vue Edit

Passer l’objet à mettre à jour dans la méthode View de Edit(Get) afin qu’il soit affiché dans la vue, sans oublier d’alimenter la liste déroulante correspondante à Plane.

// GET: FlightController/Edit/5

public ActionResult Edit(int id)

{

ViewBag.PlaneFk =

new SelectList(sp.GetMany(), "PlaneId", "Information");

return View(sf.GetById(id));

}

Mettre à jour l’objet en question dans la méthode Edit(Post)

// POST: FlightController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, Flight collection)

{

try

{

sf.Update(collection);

sf.Commit();

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

1. Créer la vue **Details** qui permet de mettre à jour un vol.

Generer la vue Details

Passer l’objet à afficher dans la méthode View de Details(Get) afin qu’il soit affiché dans la vue.

// GET: FlightController/Details/5

public ActionResult Details(int id)

{

return View(sf.GetById(id));

}

**V  Appel de service spécifique**

1. Ajouter un bouton qui permet d’ordonner les vols en utilisant l’appel au service **SortFlights()** qui retourne les vols ordonnés par Destination.

Ajouter l’action Sort dans le controlleur FlightController

// GET: FlightController

public ActionResult Sort()

{

return View(sf.SortFlights());

}

Générer la vue associée à cette action.

Ajouter le bouton Sort dans la vue Index du Flight

<**input** type="submit" value="Sort" **asp-controller**="Flight" **asp-action**="Sort" />

**V  Ajouter un lien vers un autre controlleur**

Ajouter un lien pour afficher les détails de l’avion affecté à un vol, donc Index Flight

<**a** **asp-controller**="Plane" **asp-action**="Details" **asp-route-id**="@item.PlaneFk">Plane</**a**>