# UserStories Part 8– Design Patterns

Du skal fortsætte med projektet scrumboard.

I denne opgave skal du snakke med din sideperson.

1. Forklar strukturen af mapperne og klasserne i dit ScrumBoard Projekt.

**Pages**, Her ligger alle siderne der vises i applikationen der er to niveauer  
home (roden) der indeholder Error, Home, Privacy og Index  
UserStories der indeholder CreateUserStory, DeleteUserStory, Index, UserStoryDetailed  
Læg mærke til:

Index i roden har fx URL = ” <http://localhost:8051/>”, eller ” <http://localhost:8051/Index>”  
Index i UserStories har fx URL = ” <http://localhost:8051/UserStories>”, eller ” <http://localhost:8051/UserStories/Index>”

**Services**, Her ligger de klasser der holder styr på data (her samlingen af UserStories), De ligger her og tilknyttes i klassen StartUp.cs hvor services.AddSingleton<IUserStoryService, UserStoryService> javer det vi kalder injection, dvs. laver opsætningen af servicen.

**Util,** Her ligger de hjælpe klasser som vi gør brug af fx de to klasser til at læse hhv. skrive til Json-filer.

Selve model klasen ligger i et separat projekt – et library  
**Model,** Her ligger model klassen UserStory

1. Hvad sker der når applikationen starter i en browser?  
     
   Hvis vi ikke har konfigureret projektet specielt til at ’launche’ en specielt side.  
   **Browseren** vil lave et http-get request til Web-applikation og anvende URL’en fx ” <http://localhost:8051/>”, hvor den vil aktivere OnGet i siden (page) Index.chhtml.cs, samt vise Index.cshtml (sende Index.html (den dynamiske tilpassede index.cshtml-side) tilbage til browseren som et http-response)

1. Hvornår kaldes OnGet metoden? hhv. OnPost metoden?  
   OnGet kaldes når siden (page) vises.  
   OnPost kaldes når brugeren trykker på en knap el.lign. i en form
2. Hvorledes kommer du fra en side til den næste side?  
     
   Det kan gøres på flere måder
   1. Brugeren kan i URL (browseren) skriv den ønskede side fx ”<http://localhost:8051/UserStories/Index>”
   2. I OnGet eller OnPost kan man benytte fx  
      RedirectToPage("Index");
   3. I Cshtml-filen kan du angive den nye side ved fx   
      asp-page="DeleteUserStory"
3. Hvad svarer XXX.cshtml filen til i en MVC struktur?  
   Det svarer til View altså V
4. Hvad svarer XXX.cshtml.cs filen til i en MVC struktur?

Det svarer til Controller altså C – selvom den kaldes model (Microsoft ViewModel)

1. Find og beskriv de design mønstre (Design Patterns) som findes i ScrumBoard projektet  
   Der vil sikkert være en 3-4 stykker  
     
   Blandt andet:  
   **Information expert**: UserStoryService, JsonFileReader, JsonFileWriter

**Creator**: oprettelse af UserStory i CreateUserStory

**Controller**: Alle RazorPages dvs. deres ’code behind’ eller XX.cshtml.cs

**Low Coupling**: IUserStoryService i de forskellige RazorPages

**High Cohesion**: UserStoryService, JsonFileReader, JsonFileWriter, UserStory

**Adaptor**: IUserStoryService og UserStoryService hhv. UserStoryServiceJson

**Singleton**: i StartUp service.AddSingleton<IUserStoryService, UserStoryServiceJson>()

1. Kunne du forestille dig andre design mønstre taget i brug?  
   Facade, State, evt. Strategy

Ekstra Opgave:

Du skal udvide Index siden så du kan vælge (evt. dropdown box) om User Story’ene kan vises almindelig, sorteret efter Business Value, vise de UserStories der mangler en BusinessValue hhv. StoryPoints.

1. I Controlleren Index.cshtml.cs skal der være en property, hvor valget kan gemmes.  
   I Onpost skal du sortere listen efter valget
2. Når du skal vise dem der mangler, Kan du benytte Linq-metoden \*.where( … )
3. Prøv om det virker