I architecture

Separate responsability

Each layer can elvolved independently

KitchenHeaven.Framework (Web API)

Advantage : have the same object and rules applyied, regardless of which project referenced it.

Risk : .net evolution, make it up to date to avoid .net evolutions problem/ obsolescence

DataModel :

Object that will be use in data transporation

DataAccess :

Isolate operation on data source.

Make the interface between data model and stored data

Service :

Centralize applying Business Rules.

Organize operations and call data access to perform the operations on data

KitchenHeaven.API (Web API) :

Make the intermediary beetween user request and service

Restfull => accessibility, client independent of evolutions (implementation + architecture)

Centralize Security and request validation process

Presentation :

Portable

Reduce Load time

Expect a json message from server independently to the service implementation

II Risks

nb resultat liste restaurant

filtre liste résultat

Recherche des restaurant

Recherche des menu

Recherche intuitive

extension trafic

wait from api return

exposition schema database dans model