***Final Project- Programming in internet environment***

**Presenter: Ilay zeidman, Id: 300634524**

Project in github: <https://github.com/ilayze/MvcDeganitSite>

Subject: To build a website to my wife that will organize all her recipes that scattered in many places.

Requirements: the site must contain a database that will store all the recipes.

Every recipe should have a main category for example cheese cake is in a cakes category.

In addition every recipe can have navigation properties for example cheese cake is dairy and sweet.

In the site you can find a recipe by its name or it's description .

In the site you can add new recipes, delete and update and add or remove navigation properties.

The site must be very user friendly and easy for use .

The site and all its contents should be in **Hebrew**.

**Implementation**: Database: The database will contain fix tables:

**1:** Table for recipe.

**2:** Table for main category.

**3:** Table for navigation property.

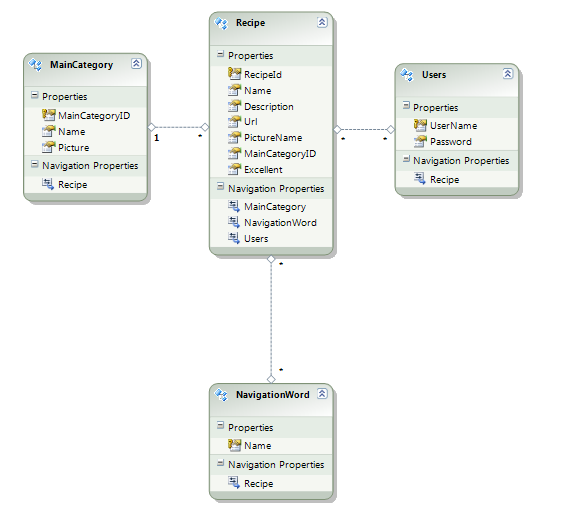
**4:** Table for user.

**5:** Table for the many to many relationship between recipe and navigation property.

**6:** Table for the many to many relationship between user and recipe.

**Database Constraints**:

* The recipe and main category tables will have a one to many relationship, every recipe have exactly one main category and every main category can have many recipe associated to it.
* The recipe and navigation property will have a many to many relationship thus will be a table RecipeNavigation that contains the keys of both.
* The user and recipe will have a many to many relationship thus will be a table UsersRecipes that contains the keys of both. Here is a class diagram illustrates this constraints:



Three levels of access:

The site is build that only administrator can change or delete existing recipes or to add main categories or navigation properties.

The first level of access is user that not logged in. User that not logged in can see the about page and the sites of recipes. After the user sign in he is in the second level of access.

The second level of access is user that logged in but not an administrator. The user has its own repository of recipes and can add recipes to its repository and can add recipes to the big repository.

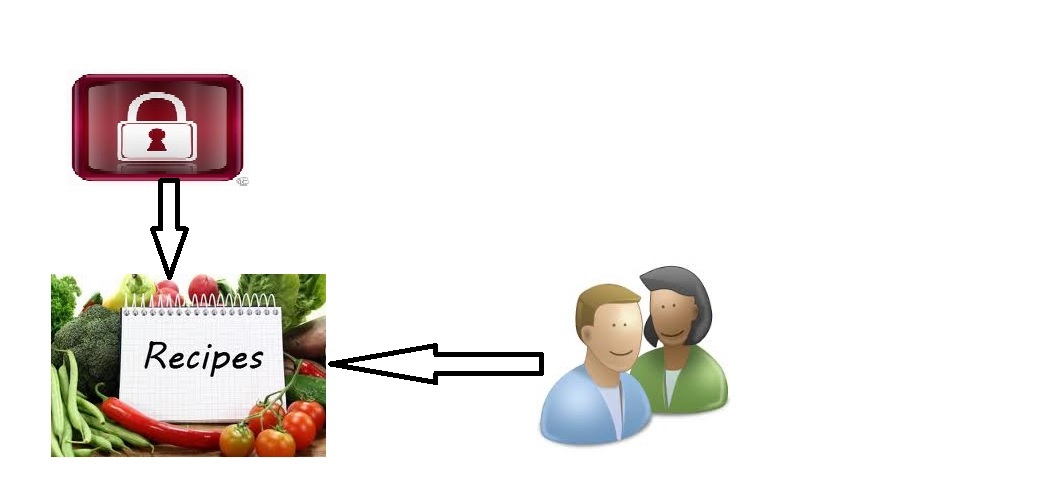
In order to add recipe to your repository go to all recipes and choose existing recipe or click on add recipe and add your own recipe.

In order to delete recipe from your own repository in the home page click on "To my recipes repository" and in the recipes repository you can delete recipes.

The third level of access is administrator, administrator can add main categories remove main categories add navigation properties, assign navigation properties to specific recipe, update delete add new recipes, mark recipe as success=> recipe that known that is good.

The following diagram shows that only administrator can change the recipes repository.

Administrator

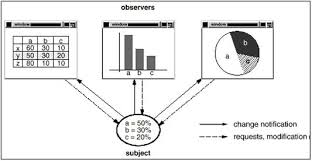


Users

Add delete and update recipes

Two views to the same model:

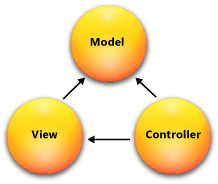
The MVC pattern separate the view from the model thus allow you to make two views to the same model as illustrated in the diagram below (In fact this is implementation of the Observer design pattern):



In the diagram you can see three views to the same model.

In my implementation I did two views to the recipes. In the menu click on "all recipes" and in the bottom of the page you can click on "switch to basic view" and you can see the same model(recipes) in two different views.

**Project**: I used ASP.NET MVC framework for the site, this framework implements the model view controller pattern.



Model: Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects retrieve and store model state in a database.

Views: Views are the components that display the application's user interface (UI).

Controller: Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction.

The pattern helps me create application that separates the different aspects of the application.

It help me specifies where each kind of logic should be located in the application.

**User guide**: run the application and you will see the home page or login page if you are not logged in.

In the home page you can press of one of the main categories in order to see your recipes in that category or to search recipes by navigation property.