# Software Design

# Dancing in the Rain

The architecture of this web site uses MVC (Model, View and Controller) software design pattern. This pattern isolates the application logic from user interface layer so the concerns can be separated. The MVC pattern can be graphically displayed as the following:

Controller

View

Model

## The Model

The model is used to manipulate the application data, including the projects and people and their relations. The model responds to the request from the view and the controller. For example, if a user clicks the create button in the web page to create a project, the model will update the databases to save the information for the new project. The model will also tell the controller that the database has been changed, and the view, the web page showing the projects, needs to be updated as well.

The databases used in this web site are MySQL.

## The View

The view is used to displaying all the data to the user. Whenever the data is changed or a user event occurred the controller will instruct the view to change the display accordingly. The view in this web application is implemented via AngularJS template.

## The Controller

The controller responds to events, including user input to the web page and the change of the data model, process the events and informs the view and the model to change accordingly.

The web site uses AngularJS to implement the controller.

The data flow of an example event

Assume a user clicked “I PLEDGE” button on the web page. Here is an example of the data flow throw the MVC system.

1. The controller will receive this event.
2. The controller will check the model layer to get the total number pledged, and add 1 to the number.
3. The controller will instruct the database to change the total pledged number in the model later to the number obtained in Step 2.
4. The controller will also instruct the view to update its web page, adding 1 to the Pledge Made label.

File Structure of the web site

* Settings.json – the project setting
* CSS – directory, holding CSS files
  + Grayscale.css – downloaded from public for some of the style of the web pages
  + Main.css – created myself for the project
* JS – directory, holding JavaScript files
  + App.js – the AngularJS files, creating the model, the controller and templates
  + Rain.js – Java Script file simulating the rain drops on the main page
* Index.html – the HTML file used to display the web pages
* Images – directoy, holding some images used by the web site
* PHP files – directory, containing miscellaneous PHP files to communicate between the database and the controller