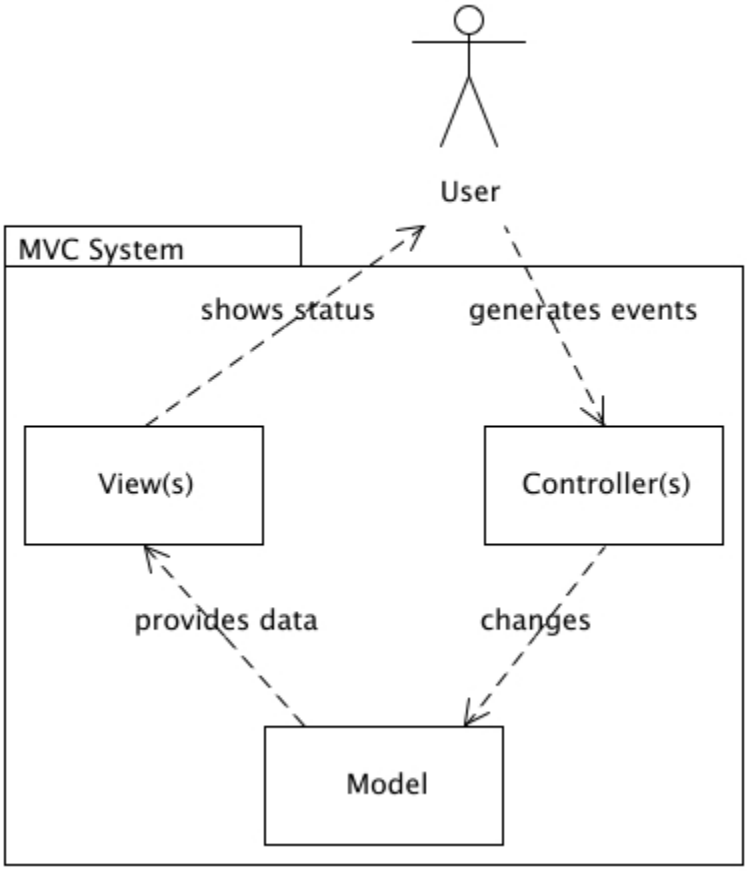
The Model/View/Controller Framework



The Startup class is responsible for creating instances of the Model & View Classes. The View class creates the ActionListeners in the Controller Class.

MVC (Model Class /View Class/Controller Class) is a way of structuring your code that defines the separation of these three types of classes:

Model objects hold data and define the logic for manipulating that data. For example, a Student object in the Basic sample application is a model object. It holds data describing facts about the object like the first and last name of the student and has methods that can access and change this data (getters & setters). Model objects are not directly displayed. They often are reusable, distributed, persistent and portable to a variety of platforms.

View objects represent something visible in the user interface, for example a panel or button. The View class creates the connection (using action listeners) between the user entry components (text field entries and buttons) and the controller code that will respond to the user action. The view obtains Model data using accessor methods.

The Controller object acts as a Mediator between the Model and View objects. A Controller object communicates data back and forth between the Model objects and the View objects. A controller performs View specific tasks, such as processing user input. The result of the response to a user input is often a a call to the Model. There is usually one controller per application or window.

Adapted from Source: http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html