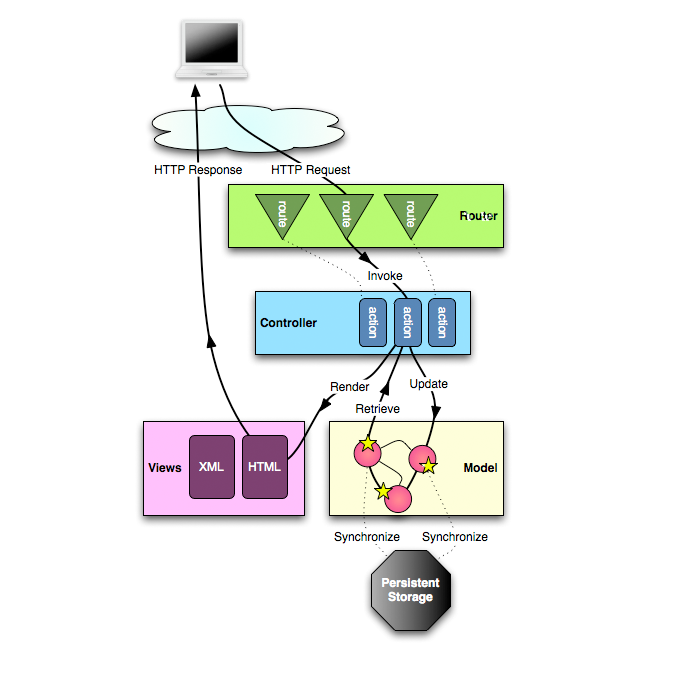
1. Architecture Overview

In this project is going to be used Play Framework. Framework splits the application into layers: Model – View – Controller. This design pattern splits data access and logic from presentation.

So, if there are some changes in user interface they will not affect any data management. And data can be handled without changing presentation layer. To make this functionality in MVC used controller component.

MVC is modular. Separation easy to support different users, and components easy to replace. Reduce complexity in large applications, increase maintainability, testability, and reusability.



**Three components of MVC:**

1. Model is directly managing Data, Logic and Rules of application. Record state of the App. Model updates View. Model links to the database. Independent of View, one Model can link to different views.
2. View (presentation layer) main functionality is to present Data to the user, it allows user interaction, but there is no any processing. It renders the model, request updates from the model. The view uses the model to generate the new user interface.
3. The controller defines application behaviour. Accept input, it is how user interface reacts to user input(event), receive messages from view, and send a message to model (tells what data to display). The controller handles user input, and update the model. Like when receiving a request for some page it loads the view of according page.



References:

<https://www.playframework.com/documentation/1.4.x/main>

<https://www.playframework.com/documentation/1.4.x/images/diagrams_path>

https://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/500px-MVC-Process.svg.png