**2) Software Architecture**

**2.1) Overview**

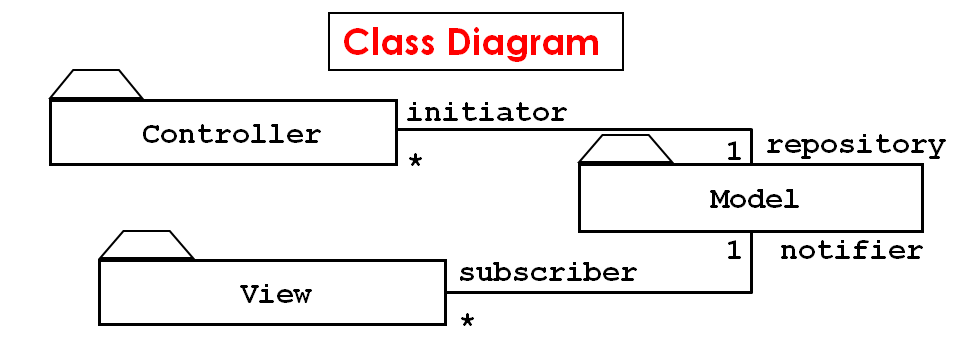
This section will be mainly on structure of the system. In the first part of the report, we have identified design goals. Now, we are modeling system design as a set of subsystems. We have decomposed the overall system into manageable parts by using the principles of cohesion and coherence. This part and third part will include identification of subsystems, services, and their association to each other. Objects and classes in the previous analysis report are ‘seeds’ for our subsystems. And uses cases in previous report allow us to define services in this part.

**2.2) Architecture Style**

We have chosen the MVC (Model-View-Controller) architectural style. In this style, we classified subsystem into three different types.

Model subsystem is responsible for application domain knowledge. This subsystem expresses the system’s behavior in terms of the application domain. View subsystem handle display related issues. Controller subsystem interacts with user, gets input and does the computations needed to be done behind and modifies both Model and View subsystems.

This style provides high coherence by allowing classes to perform similar tasks and to be related with each other, and provides low coupling by decreasing dependency between subsystems and providing that they have less information about each other. This style brings also some tradeoffs which were mentioned in the previous part.



**2.2.1) Architecture Type**

We have divided our subsystems horizontally into several independent subsystems. They have mutual knowledge about each other. While some of them have compile-time dependency, some of them have also run-time dependency. We also use open architecture in which each layer can call operations from any layer.