Class Diagram: High-Level Overview

The Class Diagram shown on the following page in figure xxxx represents a very general overview of the class structure of the CheckPoint application. The class blocks are contained within folders and these folders represent individual projects within a single solution, often but not always, a folder is indicative of a logical layer within the application. The green blocks represent interfaces and the blue blocks are classes. The high-level view that this diagram presents is unable to communicate the specific functional relationships between the classes and so this will be covered in more detail in subchapter xxxx. What this overview does make visible is the general organization of the application and the way in which interfaces have been used across the solution to enforce contractual agreements and assign interclass dependencies. Again, the precise nature of these contracts between “client” classes and “service” classes will be examined in subchapter xxxx. Dashed lines represent the implementation of an interface by a class. Solid lines represent class inheritance. The larger arrowheads point to the classes being implemented or inherited. The smaller arrowheads indicate that a class is referenced.

The folders named “CheckPointViews” and “CheckPointPresenters” both belong to the presentation layer of the application. The “CheckPointModels” and “CheckPointDataTables” folders belong to the business layer of the application while the “”CheckPointDataAccess” folder resides in the data access layer. The “CheckPointInterfaces” and “CheckPointBootstrap” folders are crosscutting concerns that transverse each of the three logical layers.



Reference figure here!