**Dashboard Code Architecture:-**

The dashboard web application is majorly divided into two sections: - **The Backend (in php )** and the **Frontend (in HTML5, jQuery, CSS3).**

**The User Interface**

All HTML is rendered on a single page (**index.php**). The Contents of this page are populated through the js file under the folder **javascript/view.**

The files under the directory **javascript/model make ajax GET request** using the gateway command object (**DashboardGatewayCommand.js**) to get data from the server

A simple Get request looks like this

**http://webnpi-beta.am.freescale.net/dashboard/service.php?service=GetBurndownChart&npi\_id=3174&burndownChartType=Errata**

Where **service.php is the controller file** for all the requests made to the server. The parameter **service specifies, which method to invoke** at the backend. Other parameters are the service specific required parameters. For example here GetBurndownChart service requires npi\_id and burndownChartType to get data.

Once data is received a **callback function** is called by ajax which in **turn populates the data** on the index.php page.

**Google’s chart api:-**

The Charts are created using google’s chart api. You need to supply a perfect data table to the function

google.visualization.arrayToDataTable(dataTable);

and then initialize it. For full API refer: <https://developers.google.com/chart/>

**Accessing the Database**

The multiple databases are accessed using a singleton mysqli object created by the file (SingletonDb.class.php).

**Reading the big Dump**

The class ParseCSV.class.php reads the bigdump csv file. It doesn’t read all the columns, it creates an array of indexes in each row to be read and reads just required columns specified by the array passed as parameter to it.

**Reading The Excel File for burndown charts**

The excel files that are uploaded to the server are read using PHPExcel API. The class ReadExcel.class.php takes in the file name in the post request and returns an iterable worksheet object.

**The CSS Framework (Bootstarp (Twitter))**

All the pop up windows(Modal windows) and styling are generated using Twitters CSS/JS framework **bootstrap**. Complete information on the twitter’s framework can be found here : <http://twitter.github.io/bootstrap/>

The files for this framework are placed under the dir css/ bootstrap.min.css and javascript/bootstrap/ bootstrap.min.js