**Business Logic Notes**

When we use the term “Libraries” we are normally referring to the classes that are located in the libraries directory and described in the Class Reference of this user guide. In this case, however, we will instead describe how you can create your own libraries within your application/libraries directory in order to maintain separation between your local resources and the global framework resources.

<?php

Defined(‘BASEPATH’)

class Someclass {

public function some\_method()

{

}

$this**->**load**->**library('someclass');

So we could use a library which contains the business logic layer to connect to the database. And through our controllers we can load the library class using the code above.

$this**->**load**->**helper('url');

$this**->**load**->**library('session');

$this**->**config**->**item('base\_url');

*// etc.*

Additiionally code igniters native resources can be called from the library using get instance()

$this**->**load**->**database();

This code is used to load the database which the class will initialise from.

Code igniter add-on which allows the user to map database tables into easy to work with objects . By using Active Record . So sql database can be mapped with this software.

http://datamapper.maglok.nl/

Can have a very thin controller and fat model which contains all the data and business logic and use that to connect to the ui view.

Queries can also be used to call specific data from the database using an array.

$query **=** $this**->**db**->**query('SELECT name, title, email FROM my\_table');

**foreach** ($query**->**result\_array() **as** $row)

{

**echo** $row['title'];

**echo** $row['name'];

**echo** $row['email'];

}

The above result\_array() function