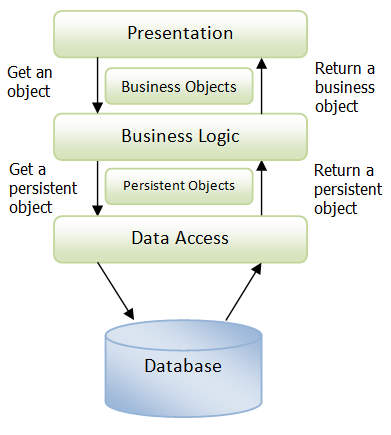
**RentBox Platform**

The RentBox platorm is a web application development framework built using in PHP. The core of the platform is implemented as a WordPress plugin. Applications built using the framework are multiple tier web application that are composed of the following tiers.

1. User Interface
2. Controller/ Request Processor
3. Application (Business Logic)
4. Model (Persistence Layer)



***N-tier application***

In the RentBox platform, the business layer is additionally separated into the Request Processor and the Business logic layers.

**User Interface**

Currently the user interface of the platform is built using HTML5, CSS, JavaScript/JQuery. The main function of the interface is to translate tasks and results into something meaningful to the end user.

**Controller/ Request Processor**

This layer coordinates the application. It process commands from the UI and invokes the relevant functions defined in the application (business logic) layer. Its is also responsible for ensuring the integrity of data to and from the GUI.

**Application (Business Logic)**

This layer encapsulates all the application specific business functionality. This layers makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the to layers surrounding it.

**Model (Persistence Layer)**

This layers is responsible for storing and retrieving data from the backend storage system. This layer is database independent in the sense that it is not tied to a specific database engine. At its foundation the persistence layer uses the PHP Laravel ORM framework to handle the actual reading and storage of data in the backend database engine.

**Data Transfer**

Apart from the exchange of data between the GUI and controller, the standard format for data exchange between components with system is via PHP arrays. Each table is modeled as Laravel model entities, which are in turn converted into PHP arrays which are then passed in between the layers. A simplified example of such an array is as follows:

*$entity\_data = array(*

*'entity\_name' => 'Property',*

*'artifact\_name' => 'property',*

*'fields' => array(…….),*

*'entity\_code' => 'SJSJDKLS',*

*'name' => 'Icon Towers',*

*'description' => 'Upscale urban apartments',*

*'address1' => ' No1 Kijabe Str',*

*'address2' => 'University',*

*………*

*);*

So to access the address attribute of a given property, one would do:

***$property\_address = $entity\_data['address1'];***