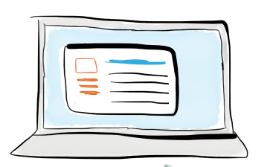
ARCHITECTURAL PATTERN

The web application was built using the Laravel framework, a modern PHP framework that aims at making PHP development easier, faster and more intuitive. The web application was built following the Model-View-Controller (MVC) architecture pattern. The MVC pattern is the separation of the user display from the control of user input and from the underlying data model or representation. The Laravel frameworks we'll be looking at all rely on the MVC, or Model-View-Controller, pattern for their code organization. Using this pattern, applications are broken into cohesive parts. Firstly, the Model layer, to handle the data, and interact with the database. Secondly, the View layer, to handle the User Interface, and lastly, the Controller, which responds to interactions going both ways. This pattern generally works well to create an application that's well-partitioned, and thus easy to test, maintain, reuse, and work on in a team environment.

2.Route to appropriate Laravel controller



3.Interact with data model

4.Controller invoke result view

1. Submit user request

database

Route

Controller

-Controller

-PasswordController

-HomeController

-AuthController

-WelcomeController

Model

-user

-account

-message

-report

-trialBalanceEntry

View

5.Render view for user

Figure 1: MVC components

When interacting with a Laravel application, a browser sends a request from the user, which is received by a web server and passed on to the Laravel routing engine. The Laravel router receives the request and redirects to the appropriate controller class method based on the routing URL pattern.

For director and secretary module the user can view report, view trial balance, see account’s information and view communication message which each other. The controller will handle user request using suitable controller whether using Controller, PasswordController, HomeController, AuthController or WelcomeController.

Controller will interact with data model to retrieve data. After invoking the model, the controller then renders the final view and returns the complete web page to the user’s browser.