

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 3, March 2019, pg.124 – 133

Design and Implementation of Web Based For Intermediate Online Shop with Laravel Framework

MOHAMMED THAKIR MAHMOOD¹; OSAMA IBRAHEM ASHOUR²;
OSMAN.UCAN³; Oğuz Bayat⁴

¹Electrical and Computer Engineering & Altinbaş University, Iraq

²Computer Engineering & Cankaya University, Iraq

³Electrical and Computer Engineering & Altinbaş University, Turkey

⁴Electrical and Computer Engineering & Altinbaş University, Turkey

¹ mohammed1991almashhadany@gmail.com; ² superstar8335@gmail.com;

³ Osman.ucan@altinbas.edu.tr; ⁴ oguz.bayat@altinbas.edu.tr

Abstract—Working with traditional methods to develop a web applications causing large limitations and a lot of time consuming and with number of unexpected errors. For this reason a new technology like MVC pattern frameworks found by some companies to deal with such issues. In this research we presented a design and implementation for web based application for e-commercial shop and third-part to buy products from online shops. We have used Laravel framework to do so. As result of this research we can find out the development was standardized and non-business logic relationships automatically processed, there was much scalability so this gives us more efficiency through the implementations.

Keywords— e-commercial; MVC patterns; PHP framework; Laravel

I. INTRODUCTION

With the broad use of Web innovation, many organizations have earnest requirements to construct their own Web business frameworks rapidly and proficiently [1]. However, an high quality application relies upon the support of very much well-designed system structure. Step by step instructions to accurately apply the center innovation to plan and fabricate a steady, adaptable and reusable Web application framework structure is the test we face [2]. Amazon, ebay, and many other online shops are so popular nowadays for online shopping and it's have been using widely by customers to buy products that don't exist in their countries or for saving their time in shopping or to buy a specific brands in that shops, so for many reasons the online shopping has been important in our life but for some reasons some online shops don't sell their products for some countries specially in the middle east for many reasons, so this is a problem need to be fixed. We will build a website to deal with this problem using a business model to buy these products for users and ship them to their homes. We will use php language for programming but by using the MVC framework because of the Variety of its

features .we will use in this document the software engineering activities and techniques to deal with problem considering the quality management through the waterfall model.

II. MVC PATTERN FRAMEWORK

The Model-View-Controller pattern nowadays is widely been used by companies and developers [3]. First time that used in Smalltalk and after that adopted and published by Java. The MVC pattern working in the following way : separates the application into three modules: Model, View and Controller. The Model is to control and to manage the data .it save and receive the entities that used by an application, usually from a database, and contains the logic implemented by the application. The View is responsible to view the data that supplied by the model in a specific format. The Controller process the Model and View layers to work together. The Controller receives a request from the client, invokes the Model to perform the requested operations and sends the data to the View. The View formats the data to be presented to the user, in a web application as a HTML output. The MVC pattern is shown in Fig. 1

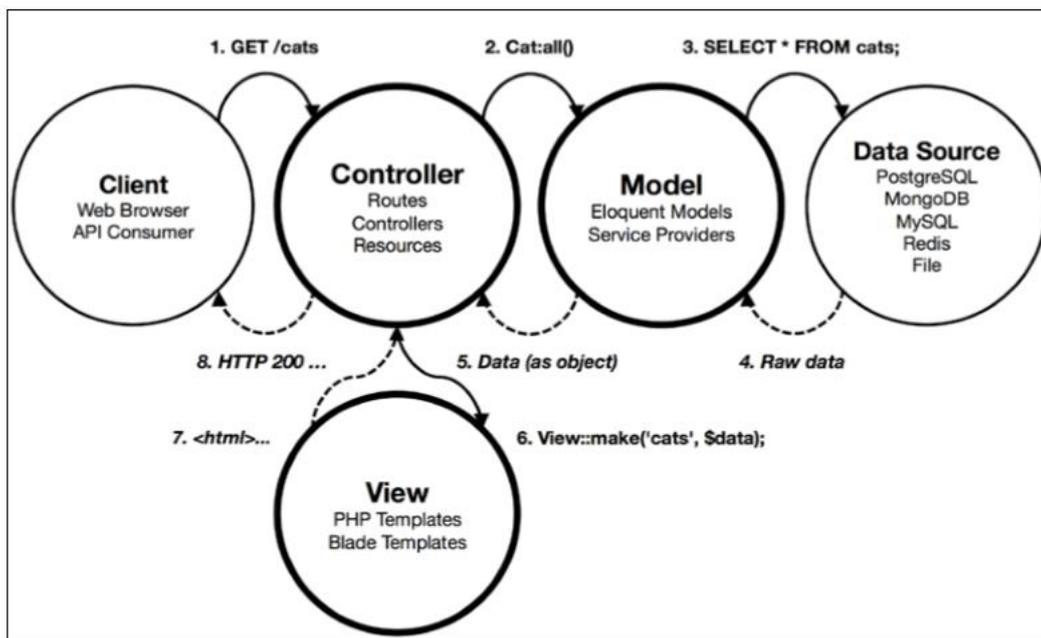


Figure 1

III. PROBLEM STATEMENT

The As we mentioned before , we are trying to design and implement a website as a third part to buy products for users that can't buy them directly . Our idea is the next, first there will be a registration part for users then they will login in their accounts. In the user profile system there will be a part to submit an order for products. In the order, there will be all information about product and the quantity of the product and the color and size (optional). After submit the order the forma of the order will received in the administration side to review it and reply to the customer or user with the acceptance or refuse on order. if the admin accept the order there will be a receipt for the order including the price information with the website's commission on the products (mostly 10% on products).The receipt will be shown to user in his profile, if the user satisfy he will pay and send a copy of the receipt for payment as an answer to buy his order .the administration will check the payment and then buy the products for user.

To build this website we are going to use the HTML and CSS with JavaScript help for the interface. For programming and data storing we are going to use php MVC framework.

IV. SIGNIFICANCE OF THE STUDY

In our study we are trying to create a website based on MVC Pattern framework. We will show in our study how we may use MVC framework works and the advantages and disadvantages of the framework, according to HE REN YU[6] he said that “Laravel make the development process is standardized ,processing some non-business logic relationship automatically. In his paper designs and implements a simple Laravel model, which achieved automated processing for part of the design. The experimental and simulation proved, web design based on Laravel framework, has scalability and robust scalability, so as to improve the developing efficiency.” And the difficulties that may any web developer may face. On the other hand the idea of website to serve customers who have a problem to buy some products from global online shops

V. GOALS OF THE STUDY

Our research focuses on building a website based on Laravel framework .we want to show to the web developers the Significance of using MVC frameworks and how to use programming languages to apply businesses models and e-commercial ideas. We believe that our website will be used by many users who have difficulties to buy some products from global online shops and on other hand get revenue from our knowledge of using programming languages.

VI. THE SYSTEM DESIGN AND ANALYSIS

Online shop web system design and planning will be presented through Unified Modelling Language UML. The design part for the application is so important for the quality management and to understand how the business are working according to the requirements[4] .Usually, the requirements are been collected by analysts and then converted into diagrams .these diagrams to show the developer how the system are working and how the data are been flowed through the system. Four UML Diagram we will be used, activity diagram, ER diagram, SQL schema Other complementary diagram is user interface. We will show some parts that may be important to be shown.

A. System Architecture

1) Customer Registration System

The most important part in the in User side is registration into the system to create customer profile. Can register to the system by this part and manage his profile. The user will be asked about his personal information like full name and phone or email and full address. After that if the information is true and there is no lack in them , the code verification will be sent via email address or phone number , the customer will be asked to enter the verification code to confirm his information for security . In figure 2 the Activity Diagram for customer registration.

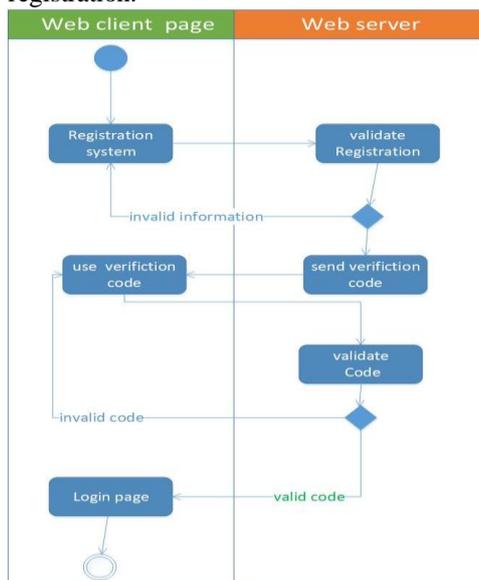


Figure 2 Registration Activity Diagram

2) Customer Profile system

User have his own profile to manage his service and buy products by add an order for products by the name of the product, link, price, quantity...etc. as we see in Figure 3. after user submit his order he will wait the administration reply on his order with invoice of total payment. The invoice contains the price of products and charging fees and commission of the website. User can pay the money in two ways, first by deposit the money in a bank and send a copy of invoice to admin to approve his order or by electronic payment with PayPal payment. In this user profile, the customer can change his information and manage his orders and he can contact with administration, he can know the price of charge for each kilogram and he can give his opinion in the website's service.

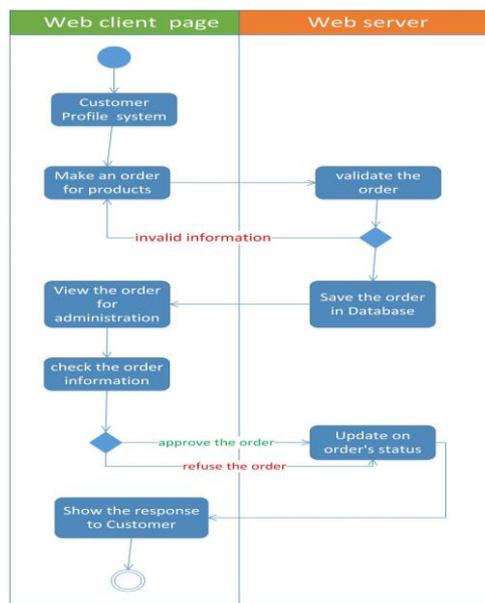


Figure 3 Activity diagram for making an order

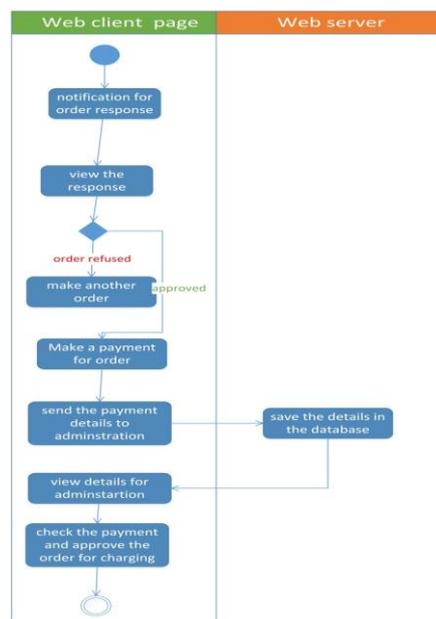


Figure 4 Activity diagram for order payment

3) Admin Login System

Admin can login to system by this part and manage his authority in the website. Every supervisor has a limited authority in the system, the Administrator who determines these authorities during the addition of a new supervisor in the system. Before the addition of supervisors the Administrator must add a supervisors groups and each group has an authority and limitations in the system. When the supervisor make a login and after validation of his accessibility in the database has been succeed, we check his group in the system and manage his authority, so if he enter a department in the admin side through his group we validate if this page is authorized within the group of supervisor in figure 8 we will explain it through the activity diagram.

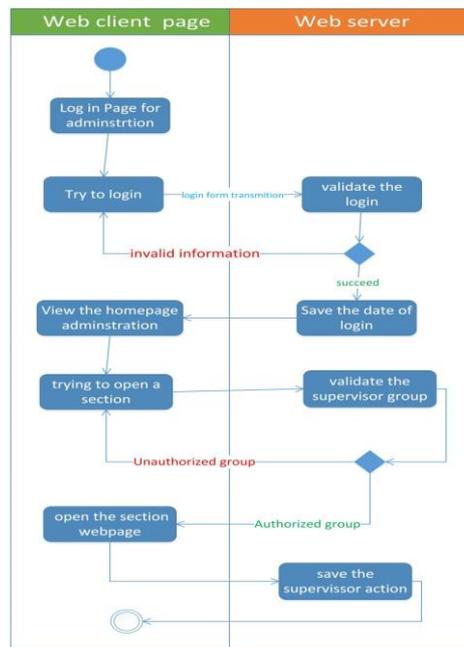


Figure 5 admin login and authority system

4) Add New Supervisors

Administrator can add supervisors and supervisors who have the authority to use this section. Administrator will provide the information of the supervisor like the name, username, phone, email and some notices and choose his authority inside the administration by choose the appropriate group. In figure 7, the activity diagram for this procedure and figure 6 for interface diagram

The screenshot shows a web form titled 'Add new supervisor'. It contains several input fields: 'First name', 'Last name', 'User name', 'password', 'Retype password', and 'email'. Below these is a 'group' label followed by a 'Select group' button. At the bottom right of the form is an 'add' button.

Figure 6 The interface diagram for add a new supervisor

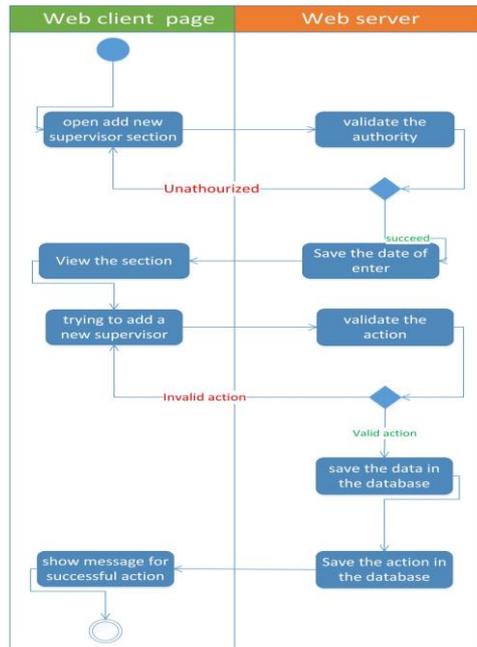


Figure 7 The activity diagram for add a new supervisor procedure

5) New Orders Section For Products Waiting To Approve

In this part admin can see the new orders for products by users and he can see the order with the details of the products like price, link, color, size. Admin can reply to user if he can buy this product and provide him with price and other details of payment through an invoice to pay it. Figure 8, the interface diagram for this section.

New orders waiting to respond						
Date of adding	Name of customer	Order's ID	Product name	State	Options	State

Figure 8

6) Orders Waiting To Confirm The Payment

In this section admin can see orders that wait for customers payment in order to buy products for them. In this part admin will see the full details of order, the payment details and the invoice of the order and if customer have paid the money he will approve the order buying and change the state of order to charged products. Figure 9 the interface diagram for this section.

Control and details page for orders waiting deposit by customer

Order's ID	customer	email	Product name	Quantity	Commission 15%	Price in dollar	Price of product in local currency	Weight in bound

Data for Customer deposit

Date	Sent Bank	Person name	Account number	Type of transfer	Received bank	Amount of money sent	Amount of money received	Notice

Data for Customer deposit by PayPal

Date	Sent account	Amount of money sent	Amount that needed to pay	Amount that sent	Notice

Invoice

Quantity	Price of product in dollar	Commission 15%	money sent	Price in local price	Price in dollar	Weight-bound

Approve Disapprove

Figure 9

B. Detailed ER Diagram
 In the figure 10 the detailed ER diagram.

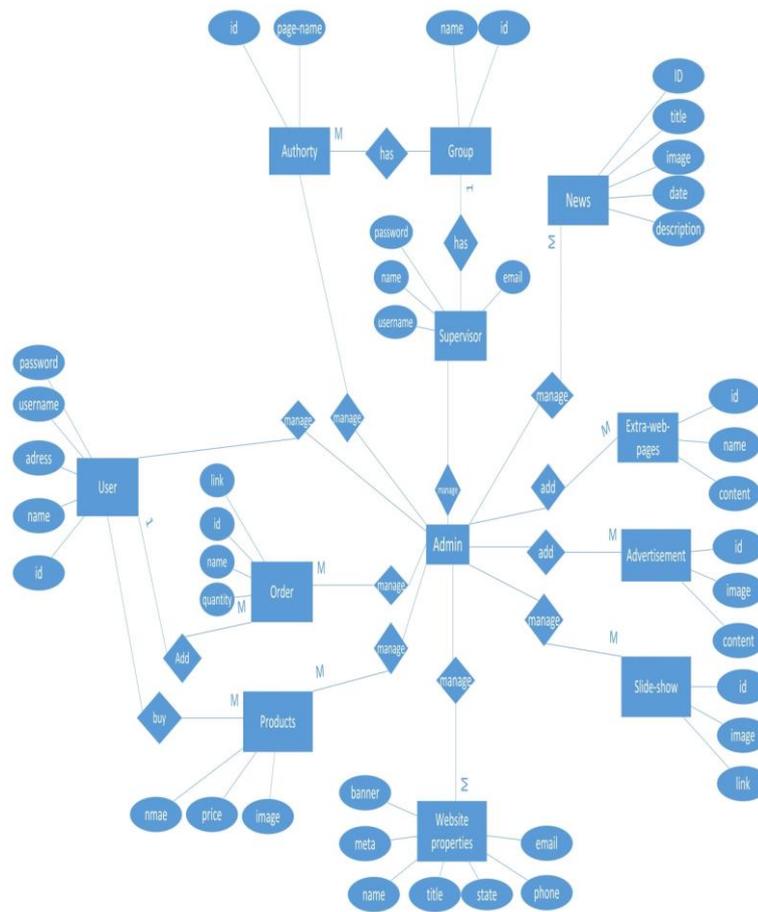


Figure 10

C. SQL Schema

In the figure 11 the SQL schema

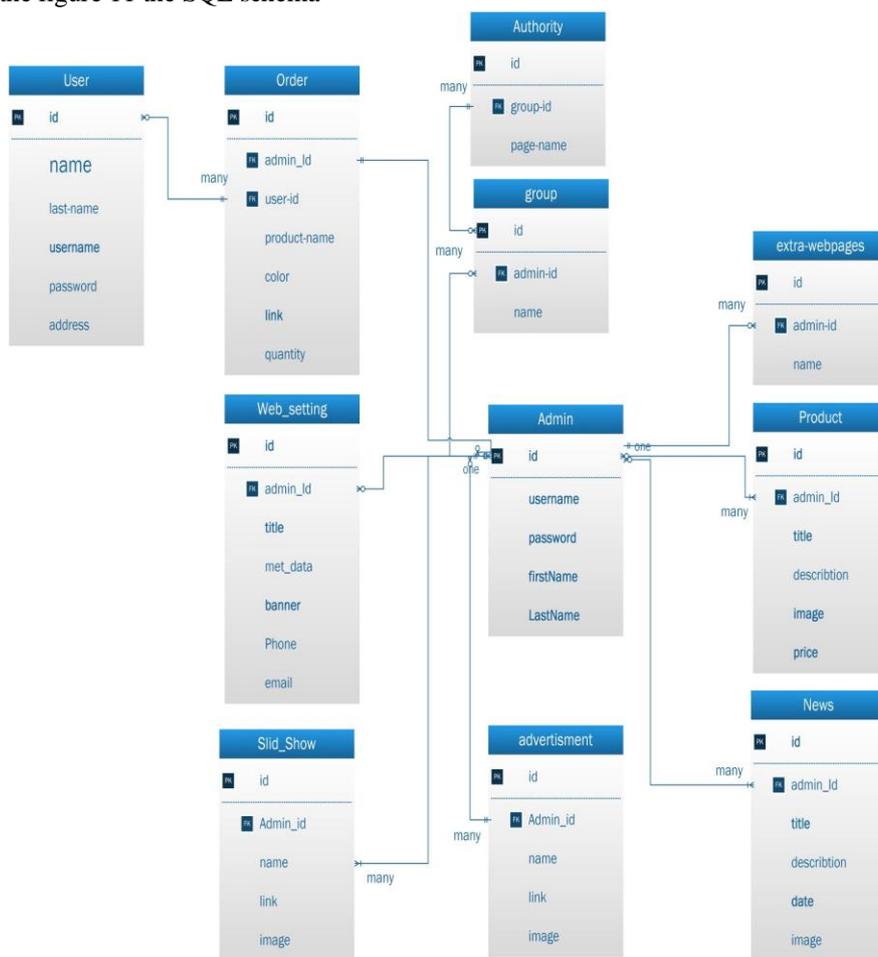


Figure 11

VII. SYSTEM IMPLEMENTATION

This is the longest phase of software development cycle where the process of writing a source code begins. The objective is to transform the gathered information from analysis and design into code. We will show some parts that could be important for this paper.

A. Creating Eloquent models

We identified two main entities: Customers and Orders. Laravel ships with Eloquent, a groundbreaking ORM that gives you a chance to characterize these substances, map them to their comparing database tables, and cooperate with them utilizing PHP techniques, instead of crude SQL. By tradition, they are written in the solitary structure; a model named Customers will guide to the Customers table in the database.

You can likewise physically characterize the name of the database table utilizing the suitably named \$table property, in the event that your table name doesn't pursue the tradition expected by Laravel: `protected $table = 'custom_table_name';` The Customer model, saved at `app/Customers.php`, will have a `hasMany` relationship with the Orders model, which is defined in the following code snippet:

```
<?php namespace Eshtry;
use Illuminate\Database\Eloquent\Model;
class Customers extends Model {
    protected $fillable = ['name','date_of_birth','Customer_id'];
    public function customer() { return $this->hasMany('App/orders'); } }
```

The \$fillable array characterizes the rundown of fields that Laravel can fill by mass task, which is an advantageous method to allot credits to a model. By tradition, the segment that Laravel will use to

observe the related model must be called Customer_id in the database. The Order display, application/Order.php, is characterized with the reverse hasMany relationship as pursues:

```
namespace App;
use Illuminate\Database\Eloquent\Model;
class orders extends Model
{
    public function users(){
        $this->belongsTo('App\user');
    }
}
```

Now after creating a model we should start with database .

B. Building The Database Schema

Now we must create the database schema., you don't have to write any SQL code and you will also be able to keep track of any schema changes in a version control system. To create your first migration, you need to use terminal window and enter the following command:

```
$ php artisan make:migration create_customers_table --create=customers
```

This code will create a new migration at database/migrations/. If you open the newly created file, you will discover some code that Laravel has produced for you. Movements dependably have an up() and down() strategy that characterizes the diagram changes while relocating up or down. Moving up is adjusting the database diagram (that is, including a table sometime in the not too distant future), while, moving down is the way toward fixing that pattern change. By tradition, the table and field names are written in snake_case. Additionally, the table names are written in plural structure.

Our customer table migration will look like this:

```
6 class CreateUsersTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('users', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('name');
18             $table->string('email');
19             $table->string('city');
20             $table->string('contry');
21             $table->string('mobile');
22             $table->string('password');
23             $table->string('address');
24             $table->string('state');
25             $table->string('forget_key');
26             $table->string('remember_token');
27             $table->timestamp('last_login');
28             $table->integer('group_id');
29             $table->timestamps();
30         });
31     }
32
33     /**
34     * Reverse the migrations.
35     *
36     * @return void
37     */
38     public function down()
39     {
40         Schema::drop('users');
41     }
42 }
43
```

C. Create a master view

Blade give us a usability to build hierarchical layouts by allowing the templates to be nested and extended . The next code screen shot is the master template that we will use for our application. We will save it as resources/views/layouts/main.blade.php. We use the master view to put all necessary files and links that we need in all pages. As we see the CSS and JavaScript files are linked and used in the master view, now we should extend these files in pages that we need by invoke the master view using the following code:

```
@extends('layouts.main')
```

This code will extend the main file and view it in the page that we invoke and this will make codes more usable and re-use these files in different pages.

VIII. CONCLUSION

The paper objectives were to analyse, design and implement a website to buy products for customers from different global shops using MVC pattern technology.

Within the Theory chapter, a brief description of used technologies such as Entity Framework, Laravel php, and so forth were introduced. The Analysis and Design part clear up the website basic features together with initial description of mentioned subcomponents. Additionally, this part presents gathered requirements based on not only customer's expectations but also the needs for an application and management requirements, also this part reflects these requirements into cohesive blocks consisting of data model, server and client side application architecture, subcomponent design and finally the application design. The last paper objective was fulfilled via Implementation part which essentially follows the same structure as previous part. Through the implementation of the system we can figure out the following points:

- The requirement gathering and design make the development process rapid and help the developers to avoid the flaws and understand the business easily.
- Using the OOP pattern make the codes more efficient and easier to reuse the functions and that make the development faster.
- Using the MVC to develop an applications is more securable because of the built classes such as authentication, routing, database managing, sending emails Class auto loading.

By the time the thesis is written, the first version of the website was successfully deployed. According to customer feedback, this solution met with success in all aspects. Thus, it can be presumed that implemented solution suits all defined requirements.

On the basis of previous information it can be claimed that all thesis objectives are considered as successfully accomplished.

REFERENCES

- [1] Yuan Jing, Cao Yaoqin, Wang Wenhai, Li Jidong. Net Based Office Automation System Based on PHP Technology [J]. *Microcomputer development*. 2003.8:61-63.
- [2] Su Chengjun, He Pilian. Applied Research on Developing Web Database with PHP [J]. *Computer engineering*, 2000.9:184-185.
- [3] M.Surguy. (2013, July). "History of Laravel PHP framework, Eloquence emerging". Maxoffsky. [Online]. Available: <http://maxoffsky.com/codeblog/history-of-laravel-php-frameworkeloquence-emerging>.
- [4] Alfat, Lathifah, Aris Triwiyatno, and R. Rizal Isnanto. "Sentinel web: Implementation of Laravel framework in web based temperature and humidity monitoring system." *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2015.
- [5] Anif, Muhammad, Arya Dentha, and H. W. S. Sindung. "Designing internship monitoring system web based with Laravel framework." *2017 IEEE International Conference on Communication, Networks and Satellite (Commnetsat)*. IEEE, 2017.
- [6] He, Ren Yu. "Design and implementation of web based on Laravel framework." *2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014)*. Atlantis Press, 2015.
- [7] Rees, Dayle. "Laravel: Code Bright." (2012).
- [8] Das, Ripunjit, and Lakshmi Prasad Saikia. "Comparison of Procedural PHP with Codeigniter and Laravel Framework." *International Journal of Current Trends in Engineering & Research* 2.6 (2016): 42-8.
- [9] Bean, Martin. *Laravel 5 essentials*. Packt Publishing Ltd, 2015.
- [10] Verma, Archit. "MVC Architecture: A Comparative Study Between Ruby on Rails and Laravel." *Indian Journal of Computer Science and Engineering (IJCSE)* 5.5 (2014): 196-198.