**A Comparative Study of Project Management System Web Application Built on**

**ASP.NET Core and Laravel MVC Frameworks**

By

Alon Poudel

A Starred Paper

Submitted to the Graduate faculty of

St. Cloud State University

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In Computer Science

October 2018

Starred Paper Committee:

Dr. Bryant A. Julstrom

Dr. Jie H. Meichsner

Mr. Channa J. Kumarage

**Table of Content**

Abstract

1. Introduction

1.1 ASP.NET Core Framework

1.2 Laravel Framework

1.3 Problem Statement

1.4 Proposed Solution

2. Literature Review

2.1 ASP.NET Core Framework

2.2 Laravel Framework

2.3 System Requirements

Table 1: The table shows the requirements of the Project Management Web Application

3. Design

Figure 1: UML Diagram MVC Design Pattern [16]

Figure 2: Project Management System

Figure 3: Architecture diagram of the Project Management System

3.1 Design of Project Management web application in Laravel and .NET Core Framework

4. Implementation

4.1 Implementation of .NET Core

4.2 Implementation of Laravel Framework in Windows under Development Environment

4.3 Implementation of Laravel Framework in Ubuntu under Production Environment

4.3 Project Management web applications in .NET Core on Windows, Laravel on Windows, and Laravel on Ubuntu

Figure 4: Login page of the Project Management System

Figure 5: Dashboard page of the Project Management System

Figure 6: Sign Up page of the Project Management System

Figure 7: Role management page of the Project Management System

Figure 8: Registered users in the Project Management System

Figure 9: Create User form in the Project Management System

Figure 10: Document page of the Project Management System

Figure 11: New Document page of the Project Management System

Figure 12: Discussion forum page of the Project Management System

5. Results and Discussions

5.1 Results

Figure 13: Page Latency of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)

Figure 14: Average Latency of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)

Figure 15: Average bytes received from the server of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)

Figure 16: Average bytes sent to the server of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)

5.2 Discussion

6. Conclusion

APPENDIX

Appendix 1:

Project Management File structure in Laravel

Figure 17: File structure of the Project Management System in Sublime text editor

Appendix 2:

Sequence Diagram in Laravel

Figure 18: User login sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.

Figure 19: Add/update documents sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.

Figure 20: Discussion forum sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.

Figure 21: Add/update notes sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.

Appendix 3

Project Management File structure in .NET Core Framework

Figure 22: File Structure of the Project Management System in .NET Core in Visual Studio

Appendix 4

.NET Core Sequential Diagram

Figure 22: User login sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.

Figure 23: Add/update documents sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.

Figure 24: Discussion Forum sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.

Figure 25: Add/Update Notes sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.

Appendix 5

Figure 26: Class Diagram of the Project Management Web Application in Dot Net Core on Windows

Figure 27: Class Diagram of the Project Management Web Application in Laravel on Windows and Ubuntu

Appendix 6

Testing of Project Management System

Figure 28: Testing of Project Management System with Jmeter

**Abstract**

With rapid advancement in the field of computer science, the way we use and interact with web applications have changed immensely. Developers must create web applications for browsers, cell phones, and search engines. That is accessible and easy to use in various devices. Therefore, the efficiency of software development is very important. Software design pattern plays a significant role to give that advantage to the developers. Software Design Pattern is an essential part of software development which is intended to solve real-world problems by creating templates of best practices. Design patterns bring in clarity, cost effectiveness and better communication in the software development cycle. It also helps with the development speed, support features, usage, and expenses. Documentation and maintenance aspect of an established web application frameworks are other attractions for developers.

The research is in Model-View-Controller (MVC) software design pattern to understand the technology, architecture, behavior, and environment. The study is a detailed analysis and comparison between ASP.NET Core and Laravel PHP web application development frameworks. MVC facilitates reuse of codes and separation of application layers. The Developer explains the development experience of Project Management Web application on .NET Core and Laravel. The web applications include a document library, a note page, and a discussion forum. It uses compatible programming languages such as HTML, JavaScript, and CSS to fulfill the web application requirements. Comparative analysis has been done based on the Developer’s experience and performance monitoring tools.

**1. Introduction**

Software design patterns are reusable solutions to software design problems that are recurrent in the application development world [1]. They are templates of formalized best practices in designing a real-world application or system.

Developing software applications is complex and costly. Programmers need to focus on the behavior and environment of each platform, technology, and architecture. Software Design Patterns brings that into the software development cycle. It helps identify potential problems before the implementation stage and reduces code complexity and code clutter.

There are various web application frameworks following numerous software design patterns. Among them, the Developer surveys the Model-View-Controller (MVC) pattern. The Developer presents analysis, design, and implementation of software applications based on MVC. A major focus is on the proper use of MVC software architectural pattern in ASP.NET Core and Laravel web development frameworks. MVC offers the possibility of code reuse and provides strict separation of concerns between the application layers. MVC is divided into three border sections: Model, View, and Controller. MVC gives more control to the developers and helps to build lightweight web applications.

Moreover, interaction with the application has changed immensely from Command Line Interface (CLI) to Graphical User Interface (GUI) making it easier for developer to control the applications. Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript has been used excessively making it easier to build a robust user interface and experience. Nevertheless, security of web applications has become an essential requirement for the development of these web applications. Due to the availability of numerous frameworks, it is essential to use the most suitable one based on application performance, learning curve, maintenance and user experience. The performance of the web application depends on various aspects such as bandwidth capacity, the number of user requests on the network, different application protocols, the application hosting environment, the programming tools and languages used to develop the application.

Open source software can be used, modified or shared for free. However, the developer needs to learn about the terms and conditions of respective open source licenses before making use of these free application frameworks to avoid any legal consequences. Quality, security, flexibility, customizability, support options, interoperability, and security open-source software, is attractive compared to propriety software solutions. Moreover, open source software can be used, modified or shared for free. There are plenty of open-source web application development frameworks available. ASP.NET Core 2.0 and Laravel MVC frameworks are a couple of examples for such open-source web application development frameworks.

In addition, it is essential to choose the smallest learning curve. This saves time and focuses on productivity. Therefore, the developer will share his experience and effort in the completion of this project.

**1.1 ASP.NET Core Framework**

ASP.NET is a scripting language developed by Microsoft for developing web applications and web pages. Moreover, ASP.NET Core MVC is a web application development framework that integrates MVC architecture and .NET platform. The framework provides the users with a clean client-side code that downloads and renders quickly in the browser. It gives stability and flexibility of rapid application development [4]. .NET Core is better for a new application that could be part of a cloud workload.

**1.2 Laravel Framework**

PHP is a scripting language that stands for Hypertext Preprocessor. PHP is free and strives to simplify web development. Laravel is a clear, simple, elegant and well-documented framework [13]. Laravel relies on industry standard conventions to reduce code. PHP framework is useful when there is an issue of time limitation, security and the complexity of the web application envisioned. Laravel has front-end, business logic, Eloquent and a database. Front page uses Blade template, jQuery, and Bootstrap for an interactive and dynamic page.

* 1. **Problem Statement**

There are many web application frameworks in the market. It is essential for developers to know about the most suitable framework for the development speed, support features, usage, and expenses. Therefore, a good understanding of the framework is vital for web developers. The focus of the research is to understand the MVC software design pattern and compare web application framework based on ASP.NET Core and Laravel.

* 1. **Proposed Solution**

ASP.NET Core and Laravel frameworks are open source and have a vast community. These are among the most popular framework. Laravel and .NET Core web applications are great for project management site consisting of a document library, a note, and a discussion forum. From this study, we will determine the advantages and challenges of using each framework.

**2. Literature Review**

Design patterns are reusable solutions to software design problems that are recurrent in the application development world [1]. Design pattern reuses successful software architectures and provides design alternatives to make a system reusable. Moreover, it helps with the documentation and maintenance of existing systems [9].

Developing generic software applications built on the dynamic web is complex and costly because programmers need to understand specific platforms, technologies, and architecture. MVC offers the possibility of code reuse and provides a strict separation between the application layers. Platforms, technologies, and dynamic web architecture produce significant cost on a technical level. Programmers are required to produce high-quality dynamic web applications within a short period [5].

**2.1 ASP.NET Core Framework**

.NET Core is better for a new application that could be part of cloud workload. It is not recommended to use it for existing workloads as it lacks some common .NET features [1]. ASP.NET Core MVC is a web application development framework from Microsoft that integrates MVC architecture. MVC is divided into three border sections: Model, View, and Controller. The View is used for the user interface and user experience. The model delivers data to the View. The Controller takes the user's request and loads the appropriate Model and View. First, the controller gets the input, and then it goes to the View. There could be many views but only one controller. This single controller selects views to be rendered. The View is unaware of the controller or a reference to the controller. However, View and Model are aware of each other as the controller passes back to the Model and not the controller serving it up. As a result, it helps manage the difficulty of large-scale projects by working on individual components [4].

**2.2 Laravel Framework**

Laravel has expressive and elegant syntax that provides a solution for development by facilitating general task in big web projects [7]. Laravel is a clear, simple, elegant and well-documented framework that focuses on equipping and enabling developers. It helps developers learn, start and develop quickly [13]. Laravel relies on industry standard conventions to reduce code. The documentation gets updated before the release of a new version to provide the latest information to the users.

**2.3 System Requirements**

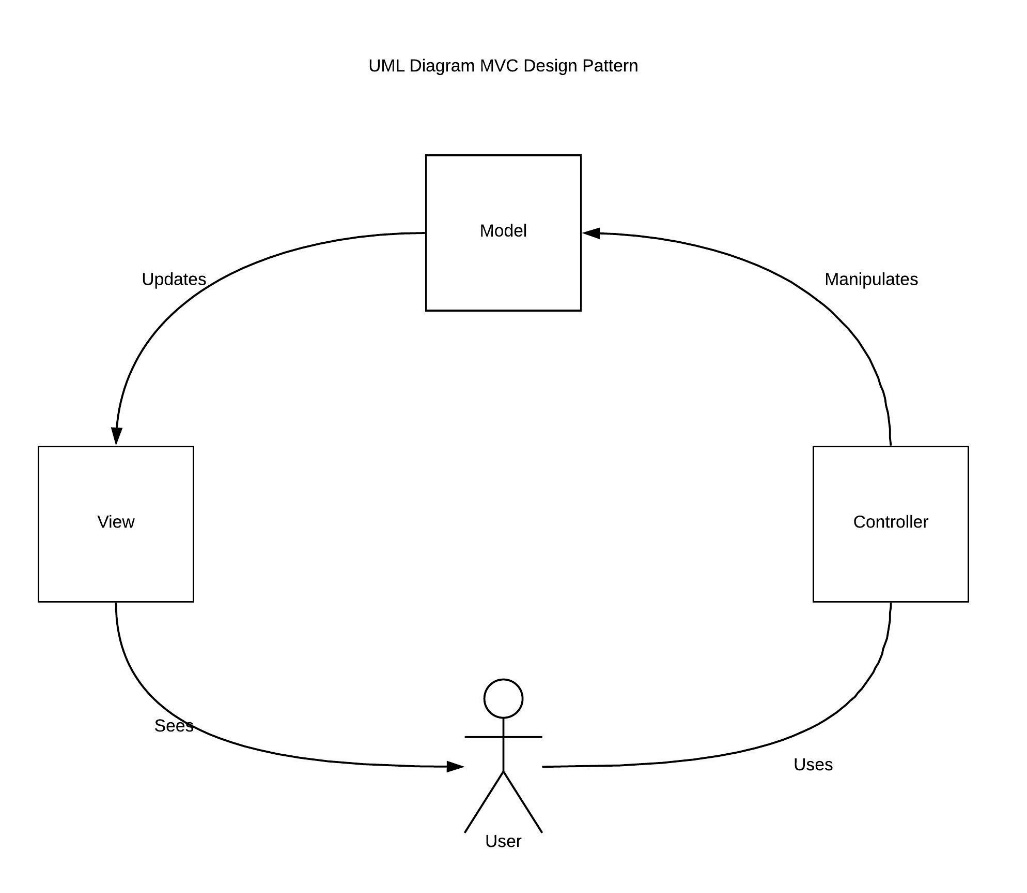
|  |  |  |  |
| --- | --- | --- | --- |
| **Components** | ASP.NET MVC .NET Core 2 (Dev) | Laravel 5.5 (Dev) | Laravel 5.5 (Deploy) |
| **Environment** | Windows 10 | Windows 10 | Ubuntu 1700 |
| **Object Relational Mapper (ORM)** | Entity Framework | Eloquent | Eloquent |
| **Template Engine** | Razor | Blade | Blade |
| **License** | MIT License | MIT License | MIT License |
| **Support and Security** | Developed and maintained by Microsoft and the .NET community on Github | Developed and maintained by Taylor Otwell and the Laravel community on Github | Developed and maintained by Taylor Otwell and the Laravel community on Github |
| **User Interface / User Experience Design** | AdminLTE 2 – Open source admin dashboard & control panel theme | AdminLTE 2 – Open source admin dashboard & control panel theme | AdminLTE 2 – Open source admin dashboard & control panel theme |
| **Programming Language** | C# | PHP | PHP |
| **Integrated Development Environment (IDE)** | Visual Studio 2017 Community | Sublime Text 3 | Sublime Text 3 |
| **Database Server** | Microsoft SQL Server 2014 | MySQL | MySQL |
| **Community** | Great Community | Growing Community | Growing Community |

**Table 1: The table shows the requirements of the Project Management Web Application**

The table above shows the components needed to develop and/or deploy Project Management System web application in .NET Core framework and Laravel framework. The mentioned list of components is also considered for comparing results and discussion. Each following column shows the type of component used to build the web application.

**3. Design**

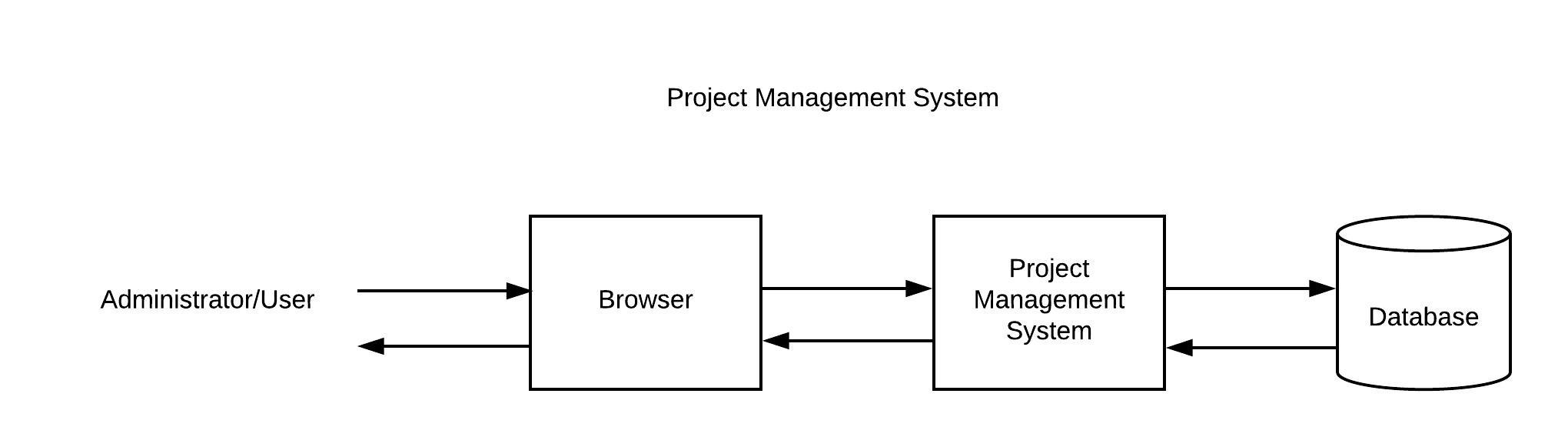
The developer surveys Model-View-Controller (MVC) design pattern in which the business logic is separated from the rest of the user interface. MVC does this by separating the application into Model, View, and Controller.MVC offers the possibility of code reuse and provides a strict separation of concerns between the application layers. The Project Management System (PMS) web application implements the MVC design pattern where the business logic is implemented in the models (M) part, which is separated from the interface (V), and controller (C) is responsible for controlling the flow of the application execution.

******

**Figure 1: UML Diagram MVC Design Pattern [16]**

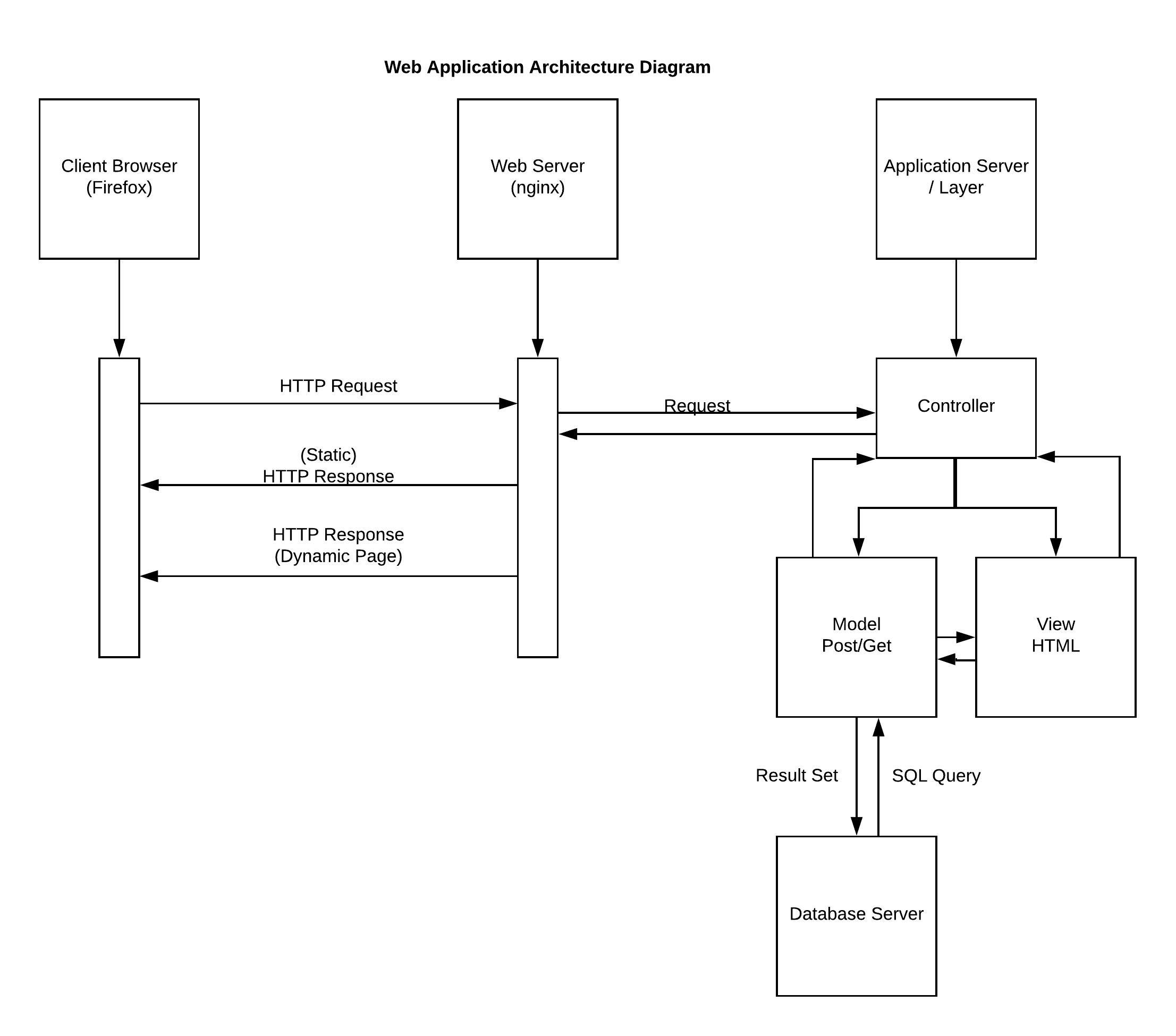
The model delivers data to the View. The Controller takes the user's request and loads the appropriate Model and View. First, the controller gets the input, and then it goes to the View. There could be many views but only one controller. This single controller selects views to be rendered. The View is unaware of the controller or a reference to the controller. However, View and Model are aware of each other as the controller passes back to the Model and not the controller serving it up. As a result, it helps manage the difficulty of large-scale projects by working on individual components while giving full control to the developers [4].

The developer designs Project Management System web application based on MVC design pattern as shown below.

****

**Figure 2: Project Management System**

Figure 2 shows the implementation of Project Management web application based on .Net Core and Laravel frameworks. Administrator/ Users logs into the system to use various features in the Project Management Site according to their permission level.

****

**Figure 3: Architecture diagram of the Project Management System**

The client browser sends an HTTP (GET/POST) request to the web server. The web server accepts the HTTP request and forwards the request to the controller of the application. The controller determines whether it needs to interact with the model or not. If interaction with model or database is not needed, it returns a view, composed of HTML markup code as HTTP response, known as static page. If the controller determines that the interaction with model is needed, it interacts with the model, which interacts with the database server using the SQL queries generated under the controller. The result set is generated by the database server which is interpreted by the controller and the controller responds with a dynamic page.

**3.1 Design of Project Management web application in Laravel and .NET Core Framework**

The Developer has developed a web application based on ASP.NET Core 2 and Laravel 5.5. The web application with the Insert, Update and Delete functionalities. The performance evaluation of web applications will include Page Load Time and the evaluation is performed using various benchmark and system performance tools.

Microsoft Windows 10 operating system is required to develop web application based on ASP.NET Core 2.0. The Relational Database Management System (RDBMS) for the web application will be Microsoft SQL Server 2014. The web server is inbuilt for ASP.NET Core 2.0. Visual Studio Community 2017 will be the Integrated Desktop Environment (IDE) or development tool to build this web application.

Laravel 5, based on PHP programming language, is hosted on Ubuntu 17.10. PHP is the programming language for the development of web application on Laravel. Moreover, MySQL is the database system. The web server is Nginx, and Sublime Text 3 is the IDE used for the development.

**4. Implementation**

**4.1 Implementation of .NET Core**

Before starting the development of Project Management System in ASP.NET Core 2.0 MVC web development framework, the developer had to setup the development environment as well as plan the development of the application.

The setup for the development involved installation of Visual Studio 2017 Community which included required packages to develop the web application, installation of Microsoft SQL Server 2014 and installation of .NET core 2.0 in the development machine.

After thoroughly planning out all the features before starting the coding, a new .NET Core ASP.NET MVC web application project was created using the Visual Studio 2017 Community edition IDE (Integrated Development Environment). To avoid the repetition of code, a generic repo and methods were added by the developer to the application.

The project was created utilizing the in-built authentication feature called ASP.NET Single Identity at first but due to its lack of flexibility in the development process, the developer had to remove the feature and build their own custom authentication module in the application. The custom authentication module involved creating of login controller where the user logins were validated, login sessions were created in the web browser if found valid and logging out of sessions to logout from the application.

The user interface framework “AdminLTE 2”, as used in the Laravel application, was used for similar appearances in all the web applications. After the user interface was integrated into the application, required changes were made to the application. Such as, changes in the login page, the dashboard, removal of unwanted files, integration of different JavaScript and CSS plugins like DataTables, and Jquery-Comments.

The developer went on to further develop the application but encountered various problems while working on different modules in the application. The design of the application showed problems such as some plugins not functioning properly, Jquery functions not working properly, and design flaws in the user interface. So, the developer had to debug and troubleshoot these problems to make it function as expected.

To help the code to be more organized, a standard was setup to be used all over the available modules. The standard was to create four different projects under the same solution in Visual Studio. The four projects were- dontnetcorepms (main project), dontnetcorepms.interfaces, dontnetcorepms.models and dontnetcorepms.repositories. They were separated based on their functionalities. The dontnetcorepms project included the main web application contents and integrated the other three projects together. The dontnetcorepms.interfaces is a class library which consists of all the interfaces needed by controllers to interact with the repositories. As the name suggests, the dontnetcorepms.interfaces includes all the interfaces which are implemented in the repositories project. The dontnetcorepms.repositories is a class library which consists of implementations of the interfaces in the interfaces class library project. The dontnetcorepms.models consists of all the models required for the project. The separation of the entire application based on their functionalities helps to debug and troubleshoot the application.

**4.2 Implementation of Laravel Framework in Windows under Development Environment**

After setting up the development environment, Laravel 5.5 was downloaded, installed and configured on the development machine.

Before starting the development of Project Management System in Laravel 5.5 PHP web development framework, the developer had to setup the development environment and plan the development of the application.

The setup for the development involved installation of XAMPP 7.0.27 which included PHP 7.0.27, Apache HTTP server 2.4 and MySQL database server. The composer and Node.js were also installed in the development machine.

After thoroughly planning out all the features before starting the coding, Laravel 5.5 was downloaded and installed on the development machine using Composer.

The installation involved configuring the application by adding a .env file which consisted the basic configuration options for the application.

Since Laravel is installed with Composer, a package dependency resolver for Laravel, it is easy to add few features needed for the rapid application development.

Composer was used to add authentication service to the application which allowed the developer to add basic authentication and a head start to build the application.

Migration service is inbuilt in Laravel, so it was enabled in the application to make it easy to migrate the application from development machine to the production machine.

An HTACCESS file along with the routing rules was added to take care of routing problem that occurred during the first phase of development of the application.

After all these, coding of the application started which involved adding new controllers, models, views, resources to the Laravel framework as per the prior planning of the application.

A third-party debugger called “Debugbar” was added to make it easy for the developer to debug the application and speed the development process of the application.

An open-source user interface template called “AdminLTE alpha” was added using npm, a default package manager for the JavaScript runtime environment Node.js.

The developer went on further to develop the application but encountered various problems working with the newly integrated interface, so the developer had to remove the npm dependency on the AdminLTE alpha user interface framework. A stable version of AdminLTE was added by the developer without the npm dependency.

The developer made changes in the composer.json file to resolve the required dependencies before issuing the Composer update command to resolve the dependencies.

The developer encountered several other issues while working on the project such as redirection problem while logging out of the application, error in the sidebar of the interface, problem in the migration of permission module, and problem in using Spatie as an authentication provider.

The developer had to rollback major changes to troubleshoot some of the problems and finally got the application to be stable and run smoothly.

**4.3 Implementation of Laravel Framework in Ubuntu under Production Environment**

After setting up the production environment in Ubuntu, Laravel 5.5 was downloaded, installed and configured on the production machine.

The setup for the production environment involved installation of nginx web server and MySQL database server. Laravel 5.5 was downloaded and installed on the production machine using Composer.

The installation involved configuring the application by adding a .env file which consisted the basic configuration options for the application.

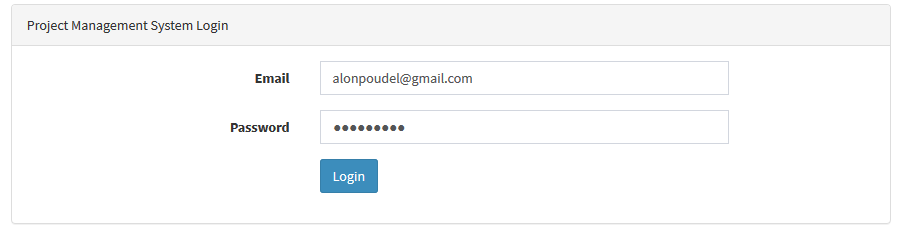
Since Laravel is installed with Composer, a package dependency resolver for Laravel, it is easy to add few features needed for the rapid application development.

Migration service is inbuilt in Laravel, so it was enabled in the application to make it easy to migrate the application from development machine to the production machine.

An HTACCESS file along with the routing rules was added to take care of routing problem that occurred during the first phase of development of the application.

**4.3 Project Management web applications in .NET Core on Windows, Laravel on Windows, and Laravel on Ubuntu**

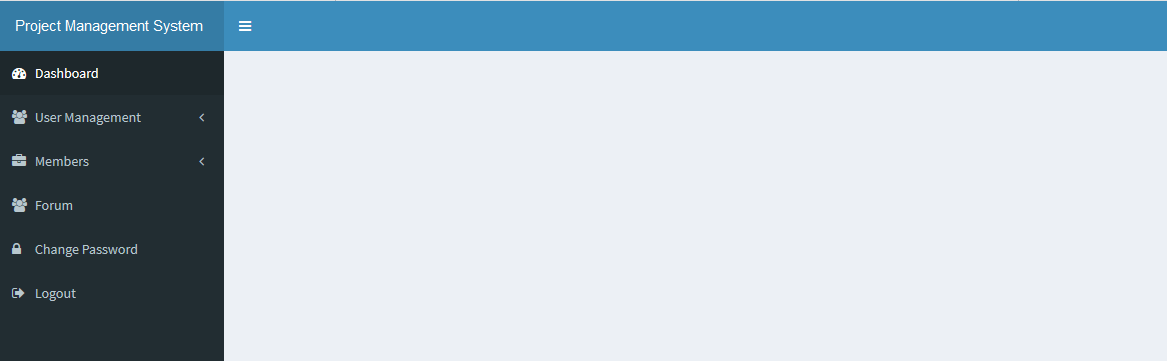
**Login Page**



**Figure 4: Login page of the Project Management System**

This is the login page for the Project Management web application in Laravel on Windows and Ubuntu, and .NET Core on Windows. This is the default page that is displayed after launching the web application. The users provide their registered email and password to access the Project Management System.

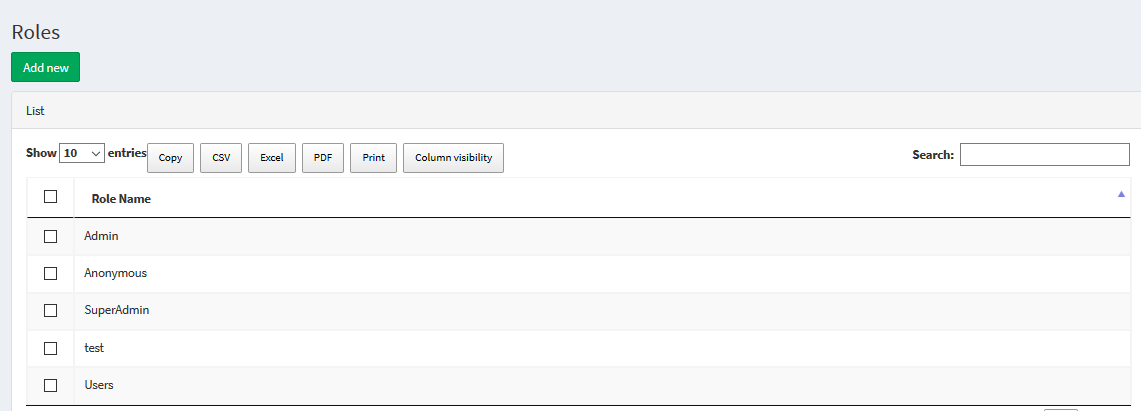
**Dashboard**



**Figure 5: Dashboard page of the Project Management System**

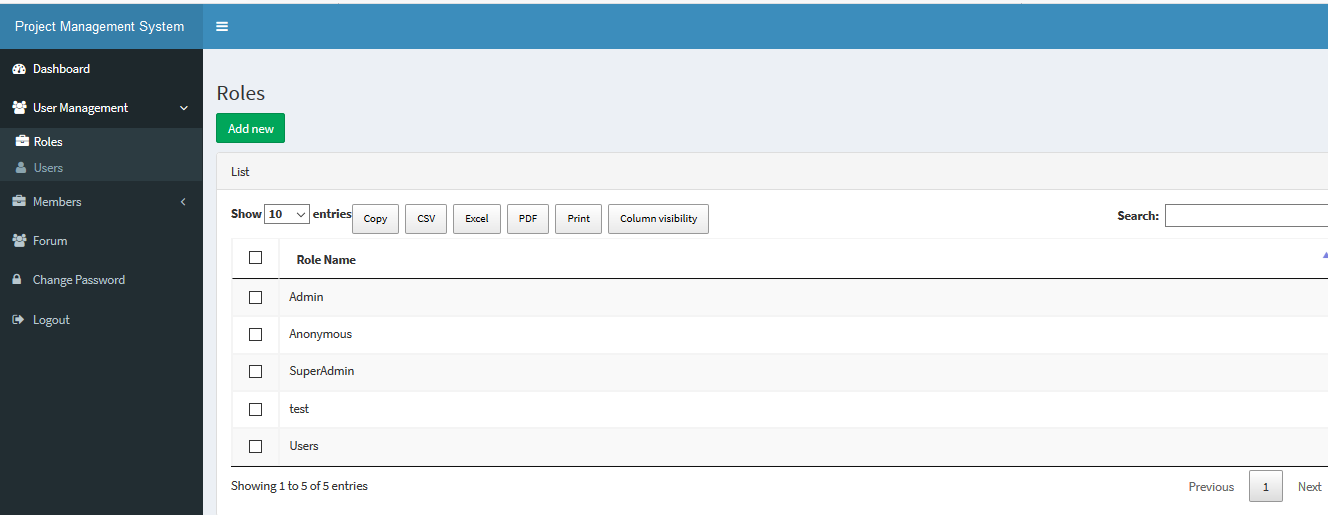
This is the dashboard page for the Project Management web application in Laravel on Windows and Ubuntu, and .NET Core on Windows. After login in, the user can access the dashboard. The left nav bar provides links to the dashboard itself, User Management (Roles and Users), Members (Notes and Documents), Forum, Change Password and Logout.

**Sign Up Page**



**Figure 6: Sign Up page of the Project Management System**

**Roles**

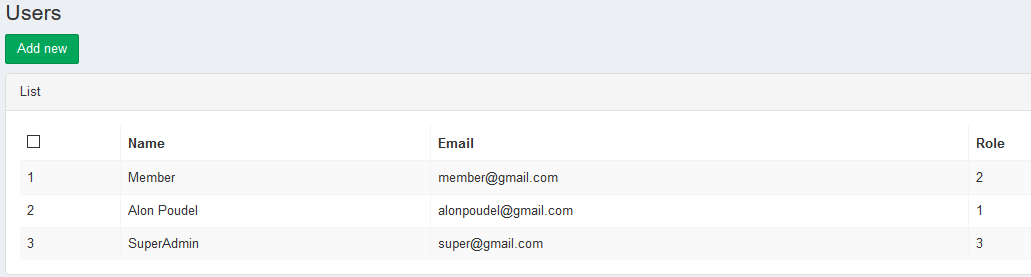


**Figure 7: Role management page of the Project Management System**

The admin assigns users according to the roles with associated permissions.

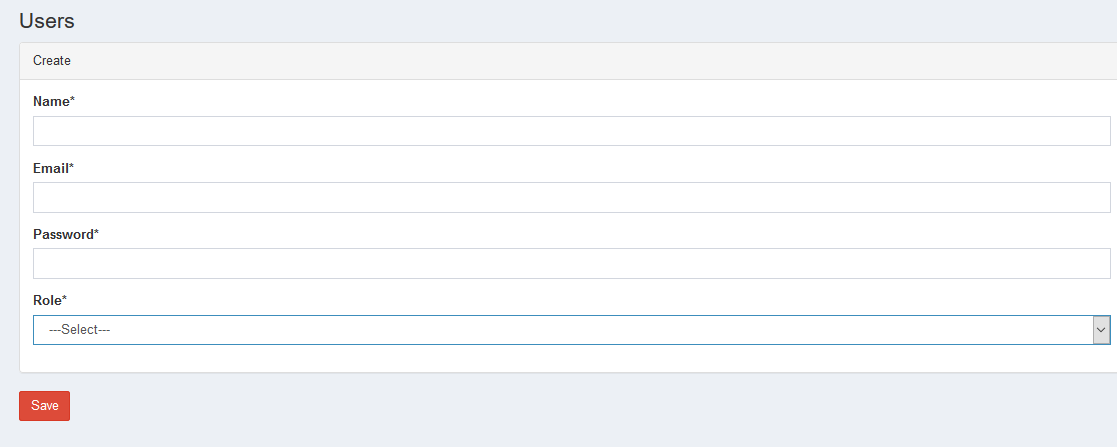
The roles are: 1. Admin, 2. Users,

**Users**



**Figure 8: Registered users in the Project Management System**

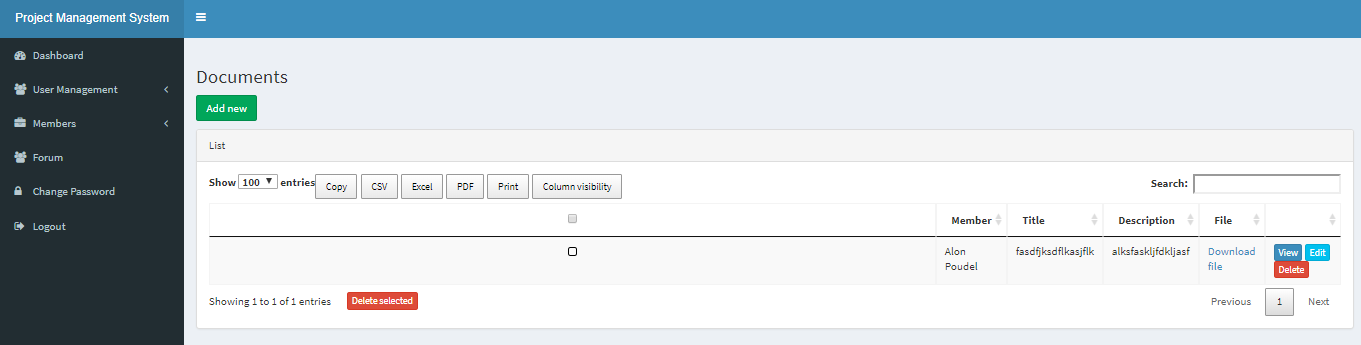
The figure shows the list of users added to the Project Management System. Only users from the list can get access to the system. The numbers from the Role column at the end is the type of access the user has. For example, the first row with “Member” is as the user name has the “User” privilege.



**Figure 9: Create User form in the Project Management System**

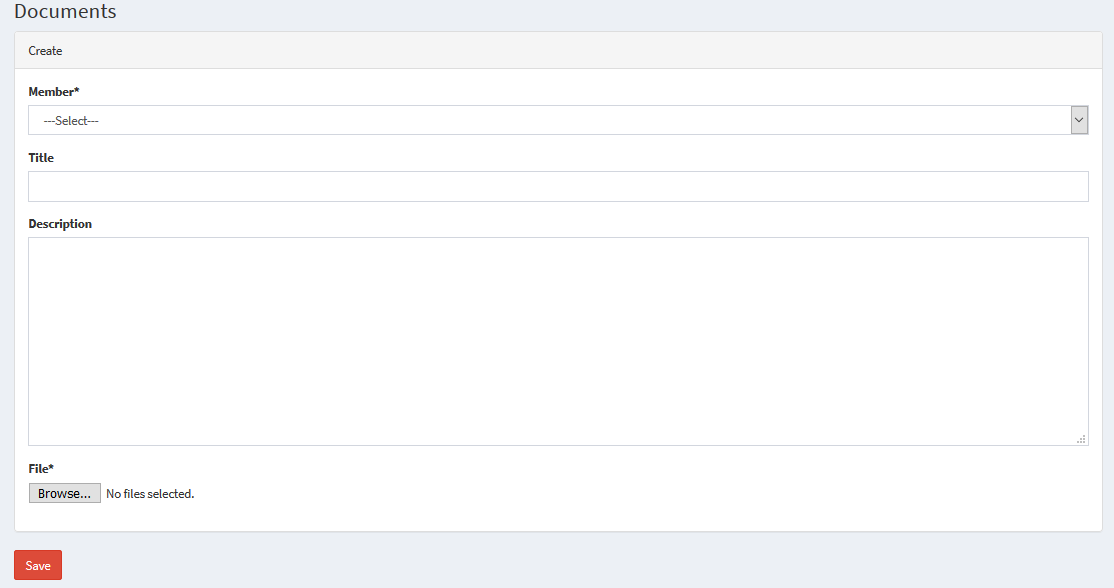
The User form gives the admin access to add new users and assign them the roles. The new user can then access the Project Management System with the assigned privilege and the password.

**Document Library Page**



**Figure 10: Document page of the Project Management System**

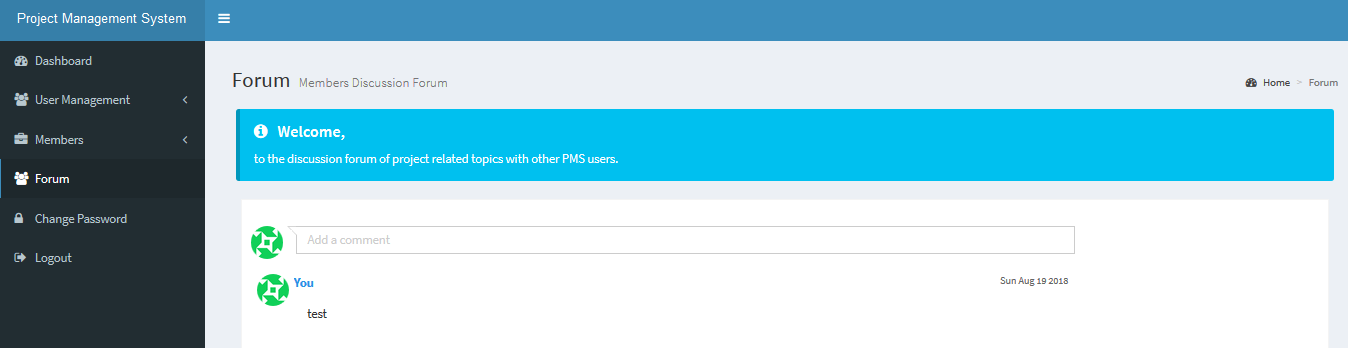
The page gives the list of documents added to the Project Management Systems. It provides the Member Name, title, description and the file information.



**Figure 11: New Document page of the Project Management System**

The new document page lets user add the new document in the Project Management System. The user can select the name, title, description and the file that needs to be uploaded.

**Discussion Forum Page**



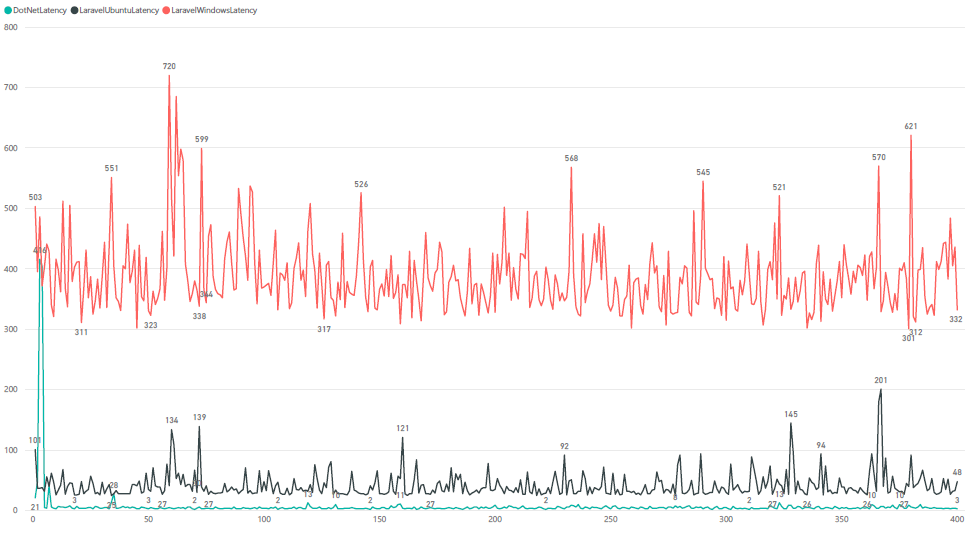
**Figure 12: Discussion forum page of the Project Management System**

The discussion forum page lets user add comments and reply to the posted comments. This is great for communicating with other users on topics of interest. Users can add comments at the empty bar with the hint “Add a comment”. To reply or edit, the user can click on the “Reply” or “Edit” button and then press “send”.

**5. Results and Discussions**

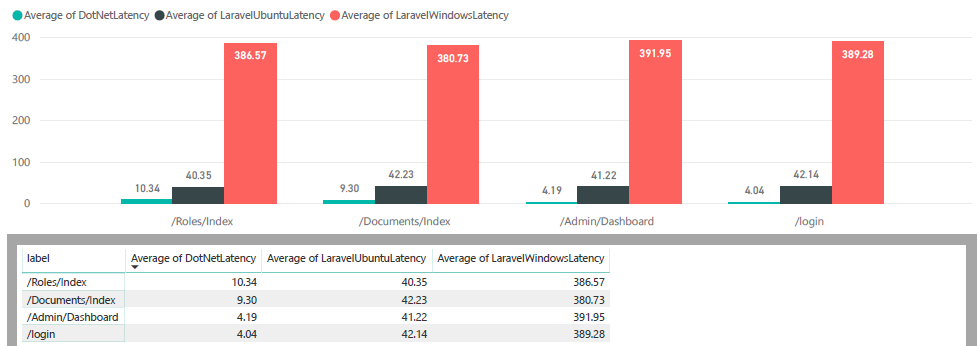
**5.1 Results**

Sample size of 400 was taken to show the comparative graph below. There are 4 pages considered for the test, with each page tested 100 times.



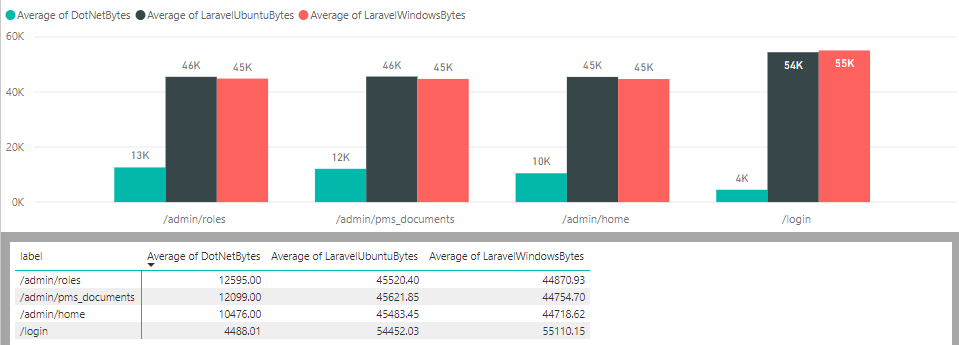
**Figure 13: Page Latency of Project Management System Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)**

Latency is the delay while connecting between the web server and the browser and vice versa. The Red graph is the latency in Laravel on Windows, Black graph is the latency in Laravel on Ubuntu and the Blue graph is the latency in Windows. The figure above presents the page latency experienced by the 4 pages, 100 times on each page, in each Project Management System web application developed in Laravel and .NET core on Windows and Ubuntu.



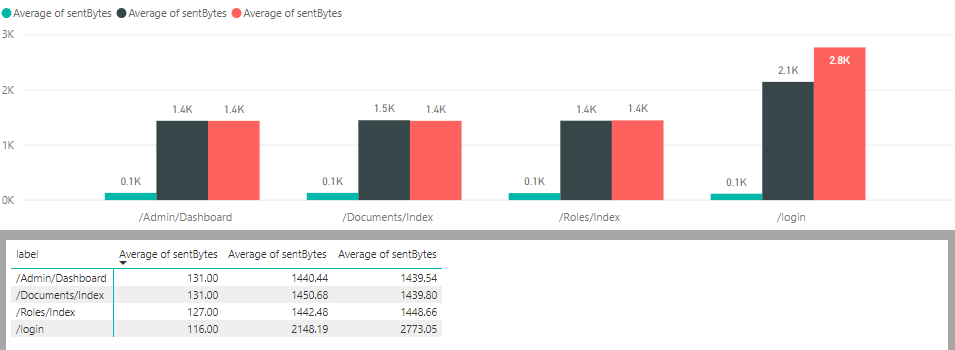
**Figure 14: Average Latency of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)**

The picture above represents the average page latency for each page on dot Net Core, Laravel on Ubuntu and Laravel on Windows respectively. Green bar is the average latency of the PMS on .NET Core on Windows, Black bar is for Laravel on Ubuntu, and the Red bar is for Laravel on Windows.



**Figure 15: Average bytes received from the server of Project Management System Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)**

Bytes represents the quantity of data in the sample response returned from the server. The picture above represents the average bytes received from the server for each page. Green bar is the average bytes received from the server of the PMS on .NET Core on Windows, Black bar is for Laravel on Ubuntu, and the Red bar is for Laravel on Windows.



**Figure 16: Average bytes sent to the server of Project Management Web Applications (.NET Core on Windows, and Laravel on Windows and Ubuntu)**

Sent bytes represents the quantity of data in the sample sent the server. The picture above represents the average bytes sent to the server for each page. Green bar is the average bytes sent to the server of the PMS on .NET Core on Windows, Black bar is for Laravel on Ubuntu, and the Red bar is for Laravel on Windows.

* 1. **Discussion**

The discussion below is based on the assumption the developer had during the beginning of the study (proposal), in comparison to the developer’s experience after development and testing of the Project Management System web application is all three environments.

***Developer’s assumption on User Interface and Experience:***

***With the rapid development in the field of Computer Science, interaction with the applications has changed immensely. Command Line Interface (CLI) was the most effective way to control computers in the past. Graphical User Interface (GUI) has made it much easier for users to control the applications.***

**Developer’s post-implementation experience on User Interface and Experience :**

In the context of web applications, scripting and markup languages like Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript has been used excessively making it easier to build a robust user interface and experience. UI refers to the way the user interacts with a device using a series of input controls and navigational components like buttons, text fields, icons, and checkboxes. And UX refers to the internal experience the user has while interacting with the various aspects of the web application. In a web application, UI elements such as input controls, navigational components, and informational components play a vital role to provide easy access and proper understanding of the application. It also facilitates the user to get what they expect from the application. The user might use the web application from a browser, cell phone, search engines, etc. It should be readily accessible and easy to use. However, it is essential to know about the limitations. In a nutshell, UI and UX play a vital role in the success or failure of the web application.

The developer has used Admin LTE 2 as the user interface for both the applications developed in ASP.NET Core MVC and Laravel 5.5. It is an open source admin dashboard & control panel theme.

Razor is a view or template engine for ASP.NET Core MVC. The developer found Razor view engine as a user friendly and easy to learn view engine, which integrates C# with HTML. The code looks cleaner using this view engine when compared to the view without one. Blade is a simple, easy to learn, and powerful view template engine for Laravel web development framework. But, the developer found Blade to be less user friendly and less clean in comparison to Razor. The Razor is integrated into the Visual Studio. It is easy to maintain a clean look of the Razor markup code. The Visual Studio auto indents the code for the clean look. The Visual Studio also allows auto completion of the code. Whereas, Sublime Text does not provide auto indention and completion feature like the Visual Studio.

***Developer’s assumption on Security:***

***Security of web applications has become an essential requirement for the development of these web applications. With the rapid increase in the development of various software and applications, security is becoming a subject of grave concern.***

**Developer’s post-implementation experience of Security:**

Let us assume that the data travel from a point to another point on the Internet. The data may need to move through various points in between the source to destination. It makes the data vulnerable to data breach and interception or data alteration by unauthorized or malicious users. These vulnerabilities can create a severe threat to the information security. Therefore, the framework should enforce various security measures to ensure that the web application is secure. There should also be a limitation on the features of the web application for better security. Going beyond its capability will bring in various problems like hacking, network congestion, and attacks.

The support and security vulnerabilities of .NET Core are addressed by Microsoft and .NET community on Github. If the .NET Core has any security vulnerabilities or any software related issues, it is more likely to get fixed immediately. Unlike .NET Core, if any security vulnerabilities are discovered in Laravel, the issues are addressed by its developer Taylor Otwell and the Laravel Github community. Large organizations are more likely to choose .NET Core over Laravel as their web application development framework because Microsoft is a trusted name and can provide enterprise level support to the .NET Core framework.

***Developer’s assumption on Application Performance:***

***Web application performance is the measurement of quality and efficiency at which the web application performs. The application performance indicator can help the developers to make the web application more robust and minimize the response time of the web application.***

**Developer’s post-implementation experience on Application Performance:**

The performance of the web application depends on various aspects such as bandwidth capacity, the number of user requests on the network, different application protocols, the application hosting environment, the programming tools and languages used to develop the application. For proper application performance, the pages should render efficiently with less browser connection and content. The web application and portals need to be fast and flexible. Long page load time influences users to move on to the next website instead. It should focus on easily scalable web application framework.

Project Management System in Dot Net Core on Windows is the best performing web application when compared in terms of average latency, average bytes received from the server, and average bytes sent to the server. Then the web application in Laravel on Ubuntu is slightly better than the web application in Laravel on Windows in all the mentioned criteria. Therefore, the developer would recommend .NET Core on Windows as the preferred framework in terms of web application performance.

***Developer’s assumption on Technology:***

***Web application development comprises of two categories: client-side programming and server-side programming. Client-side programming, also known as front-end programming, includes everything that a user sees on a web browser. Whereas server-side programming, also known as back-end programming, deals with the logic and functioning of web applications. Before choosing a technology to develop a web application, a developer needs to have proper insight into the process of web application development. As there are plenty of tools and technology to choose from, the developer must take different aspects such as performance, user interface, scalability, software design pattern, web frameworks, application protocols, servers, database, and storage into account before opting for the right technology for the development of web application. Hence, choosing the right technology to address these aspects is very essential.***

**Developer’s post-implementation experience of Technology:**

The developer needs to understand different types of protocols, mark up and scripting languages. Scripting languages such as HTTP, JavaScript, cascading style sheets, and other resources which are used by the application. It is also important to understand their impact on performance concerning a specific web application framework. It is only logical to avoid creating something that is already there and improved upon rigorously. Choosing a framework that a development team is comfortable with is a significant advantage. Accordingly, the developer focused on Laravel, .NET Core, SQL, MySQL, Object Relational Mapper (ORM), Entity Framework, Eloquent Framework, PhpMyAdmin, Xampp, nginx and Admin LTE.

The Object Relational Mapper (ORM) is a tool or technique that allows conversion of data between incompatible type systems using an object-oriented paradigm. The .NET Core uses Entity Framework as an ORM. The Entity Framework (Core) is not available in ASP.NET MVC project (if a blank project is selected) but can be installed through the nuget package manager. Laravel 5.5 uses Eloquent as an ORM and can be found in-built within the framework. .NET framework supports Entity Framework integrated with EDMX database model. This allows to import database and its components directly from the database server. However, this cannot be found in .NET core. The developer had to write the code for the database context, database set, transient service, and models. It would have been easier and faster if the EDMX database model was available for .NET Core. In the context of Laravel, Eloquent handles most of the database connection and database components easily. The developer only had to write the model for the database to interact the Laravel application with the database server.

The web application developed using ASP.NET Core MVC utilizes C# as the programming language. The web application developed using Laravel 5.5 utilizes PHP as the programming language. Both programming languages are general-purpose programming languages and have their own advantages and disadvantages. However, the developer has found PHP to be easier to learn and get mastery over than C#. On the contrary, C# is more preferred language to develop enterprise applications than PHP.

The ASP.NET Core MVC based web application uses Microsoft SQL Server 2014 as the database management system (DBMS) server and the Laravel based web application uses MySQL as the DBMS server. The MSSQL Server has SQL Management Studio as the graphical interface to manage the database. Whereas, PhpMyAdmin has been used as the graphical interface for MySQL database server. The developer preferred MSSQL over MySQL based on their functionalities, performance, features and integration.

***Developer’s assumption on License:***

***Open source software can be used, modified or shared for free. However, the developer needs to learn about the terms and conditions of respective open source licenses before making use of these free application frameworks to avoid any legal consequences.***

**Developer’s post-implementation experience of License:**

Quality, security, flexibility, customizability, support options, interoperability, and security open-source software, is attractive compared to propriety software solutions. Moreover, open source software can be used, modified or shared for free. There are plenty of open-source web application development frameworks available. ASP.NET Core 2.0 and Laravel MVC frameworks are a couple of examples for such open-source web application development frameworks. Even though these application frameworks are free to use, the developer needs to learn about the terms and conditions of respective open source licenses before making use of these free application frameworks to avoid any legal consequences. Hence, it is significant to research on licenses of the web application frameworks and its add-ons, plug-ins, and extensions.

.NET Core is the open-source version of popular .NET framework developed by Microsoft. Both .NET Core and Laravel are licensed under the MIT License. The license allows any developer to use the frameworks for commercial purpose, modify them as per their need, distribute them, or any private usage.

***Developer’s assumption on testing:***

***The testing is done to increase the quality, reusability, efficiency, and software bug fixes in the code. The term unit testing refers to a practice in software development cycle where each component of the software is tested for any bugs or defects.***

**Developer’s post-implementation experience on Testing:**

The testing is performed to increase the quality, reusability, efficiency, and software bug fixes in the code. The primary purpose of testing is to ensure that each unit or function or area of the software performs as expected. A well-tested web application that behaves and performs as expected will only enhance overall user satisfaction. Moreover, it will increase the chance of new features working correctly without any unexpected output. It is great to have the capacity of application unit. In terms of testing, Project Management System web application built in .NET Core and deployed on Windows performed far better than the Project Management System in Laravel on Windows and in Laravel on Windows. The developer recommends Laravel on Ubuntu based on the results shown above.

***Developer’s assumption on the Learning Curve: It is essential to choose the smallest learning curve. This saves time and focuses on productivity. Therefore, the developer will share his experience and effort in the completion of this project.***

**Developer’s post-implementation Learning Curve and Experience:** The developer has used Visual Studio 2017 Community as an IDE for the development of web application using ASP.NET Core MVC. The developer has used Sublime Text 3 for the development of web application using Laravel 5.5. The developer has found Visual Studio 2017 Community to be more advanced, more useful, helpful and user-friendly tool when it comes to developing .NET core applications. However, it takes more time to install and takes more disk space on the development machine in comparison to Sublime Text 3. On the other hand, Sublime Text 3 is light weight and fast IDE for Laravel.

***Developer’s assumption on the Community: Active community support is the lifeline of any design pattern. Diverse and highly motivated members are the signs of a great community.***

**Developer’s post-implementation experience of The Community:** Choosing a framework with a friendly community ensures the web application’s longevity. Proper documentation of the web application framework is essential for the web application’s growth and stability. The web application framework backed by the active community will ensure regular updates, application robustness, and high performance.

Both .NET Core and Laravel have good community presence. However, .NET Core has greater community in comparison to Laravel. The developer has found that .NET Core has greater community on Github than Laravel, but Laravel’s community is also growing faster in comparison.

**6. Conclusion**

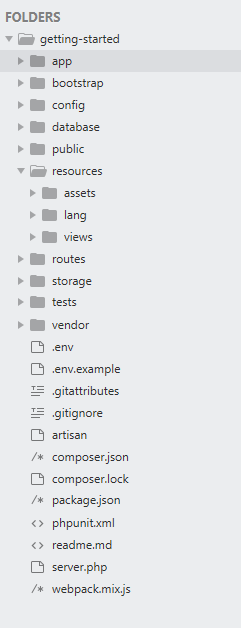
In conclusion, the learning curve, associated features and the application performance of the Project Management web application build in .NET Core on Windows is the best choice for the developer. The developer felt comfortable setting up the environment and building the application in .NET core on Windows. While Laravel was easier in few aspects, the developer had to spend a lot of time to set up the environment and building the application. Nevertheless, PHP language was friendly and got easier with time.

Moreover, the developer learned about the Model-View-Controller (MVC) design pattern, .NET Core framework and Laravel framework. These are very important concepts and software knowledges that will help the developer to further learn and further contribute in the field of computer science as an academic and a professional developer.

**APPENDIX**

**Appendix 1:**

**Project Management File structure in Laravel**



**Figure 17: File structure of the Project Management System in Sublime text editor**

**Page Description of Project Management application in Laravel Framework**

Description of important pages in the Project Management System web application in Laravel framework are listed below:

**PMSDocumentController.php:** The PMSDocumentController is responsible for adding the document to the Project Management System (PMS) by the user where the document is shared with other users to view and download. The document may be a PDF, or Text file and it can be only edited and deleted by the user or owner of the document.

**PMSForumController.php:** The PMSForumController is responsible for providing common platform for all the users or members in the PMS to discuss on a common topic.

**PMSNoteController.php:** The PMSNote is responsible for adding notes so that the user can add his/her note and share the note among all the members in the PMS.

**RolesController.php:** The role controller is responsible to manage roles in the PMS. The role can be either a super user (administrator) or a normal user where the super user has full privileges to add, delete, edit and read unlike the normal user who has limited privileges throughout the PMS.

**UsersController.php:** The UsersController is responsible for managing the users in the PMS. It can be accessed only with administrator/super user privileges. It can be used to create new user, delete the existing user, update the information of the existing user or list out the available users in the PMS.

**UploadFileTrait.php:** This is a helper file in Laravel to help the application to upload the file into the application server. This is responsible to create a new folder if it doesn’t exist and upload the user uploaded file based on the file type. For instance, if the user uploaded file is an image file, it resizes the file size and uploads into the server. It is also responsible for managing the file permission so that it is accessible to the application users.

**Controller.php:** Controller is the base controller class which provides few convenience methods such as the middleware methods, which may be used to attach middleware to controller actions. Simply put, it is a basis for providing good functionalities to newly created controllers.

**StorePMSDocumentsRequest.php**: StorePMSDocumentsRequest validates the form data obtained from the form submitted by the user. It validates the data based on the rules written in the file. It successfully executes the database operation if the validation is successful but denies the database operation if the given rules are not validated. For instance, if a form field requires numeric data to be filled but the user supplies special character then the application will issue a warning that the input data is not valid. This applies to all the files under Requests/Admin folder.

**PMSDocument.php**: This file consists of PMSDocument class which inherits its functionalities from the Model class. PMSDocument consists of field names such as name, description, file, user\_id which are used to interact with the corresponding field names in the database using the Eloquent ORM in Laravel. These field names are bound in the forms which in turn carry the data from the user and interact with the database. This applies to all the models

**App.php:** This file is responsible for the basic configuration of the application. The properties of the application such as the application name, application environment, debugging settings, application URL, application providers, aliases, etc. reside here.

**Database.php:** It consists of database connection settings for the application. Such as name and type of the database to connect, user credentials of the database to connect, etc.

**PMSDocumentFactory.php**: It is a feature that comes with Laravel which allows to create fake data for models. It is very useful to test the application and seed the database with the fake data in order to test the application in action before it is made available for real-time users to use. This applies to all files under Factories.

**Migration:** Migration is a feature in Laravel that allows creation of tables in the database. For instance, during the deployment of the application, it won’t be necessary to create tables manually in the database, migration in Laravel will take care of creating those required tables into the database.

**Seed:** Seed allows to insert default data into the tables in the database. It populates tables with default data in the database after the tables are migrated.

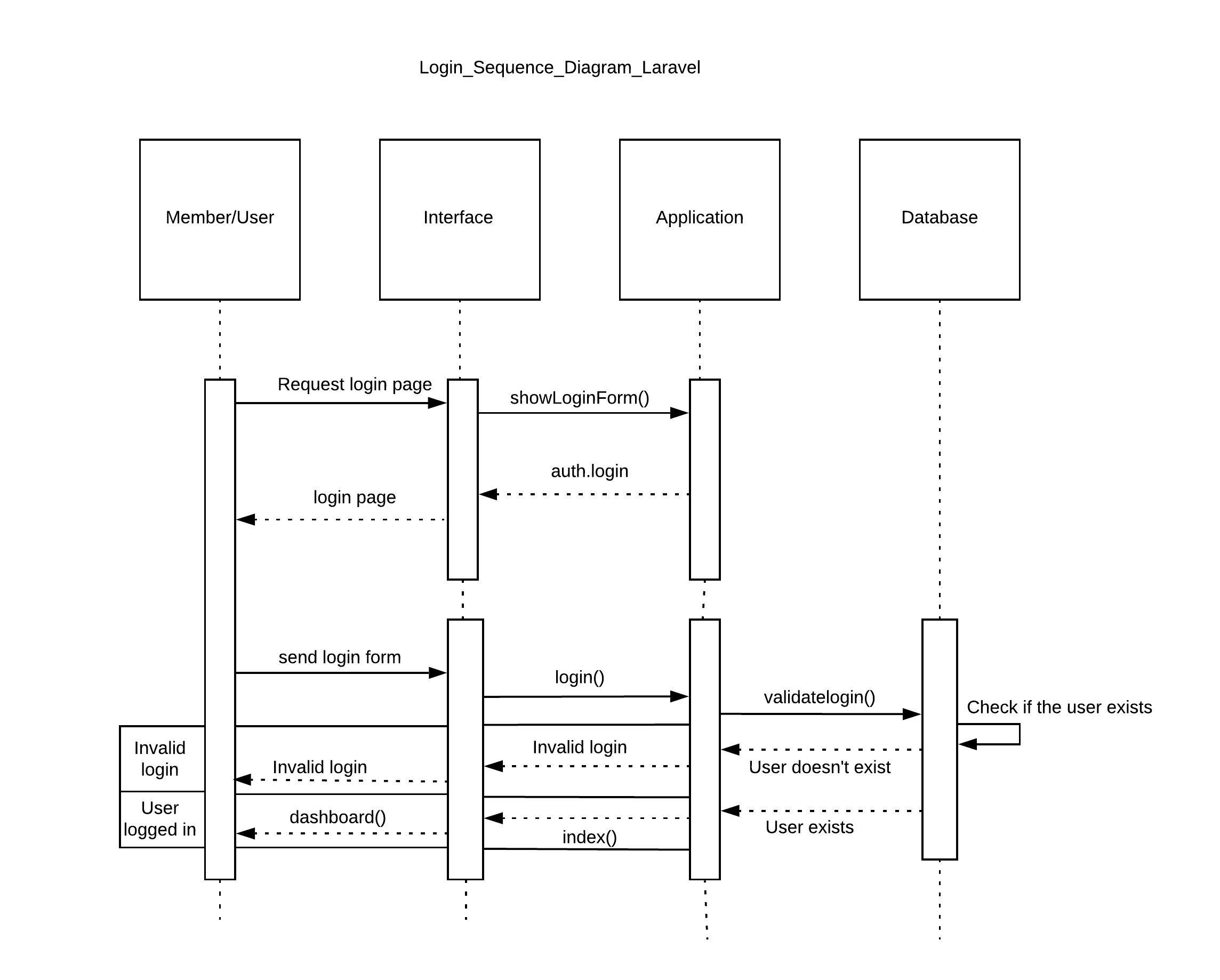
**Public folder:** The public folder consists of web resources such as css, images, javascript files of the application.

**Resources/Views:** The views folder consists of files which handle the user interface or presentation part of the application. The controller calls the view method and renders the layout in these files.

**.env:** .env is the configuration file for Laravel which consists of important variable names and their corresponding values which are used by the application.

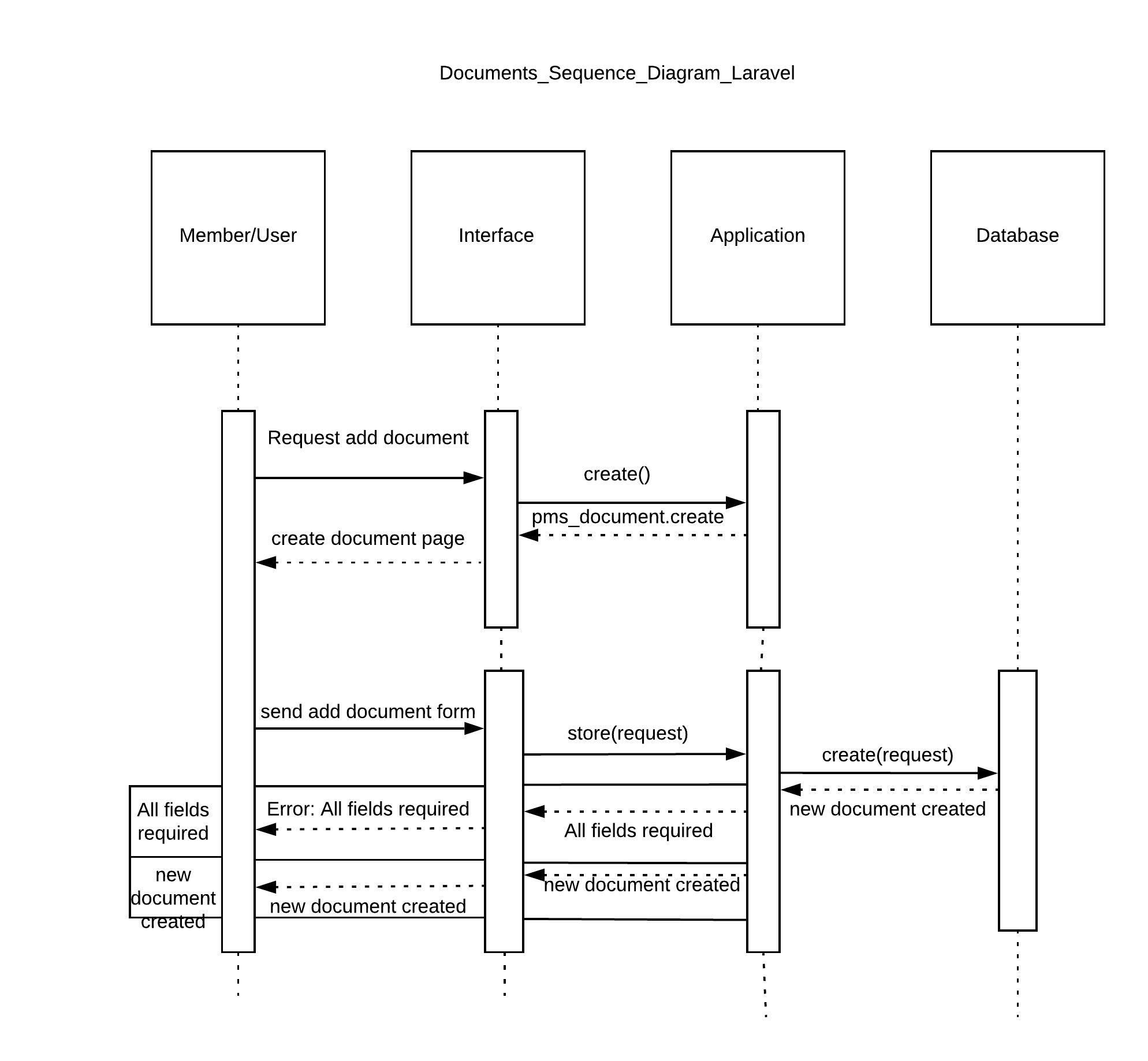
**Appendix 2:**

**Sequence Diagram in Laravel**

****

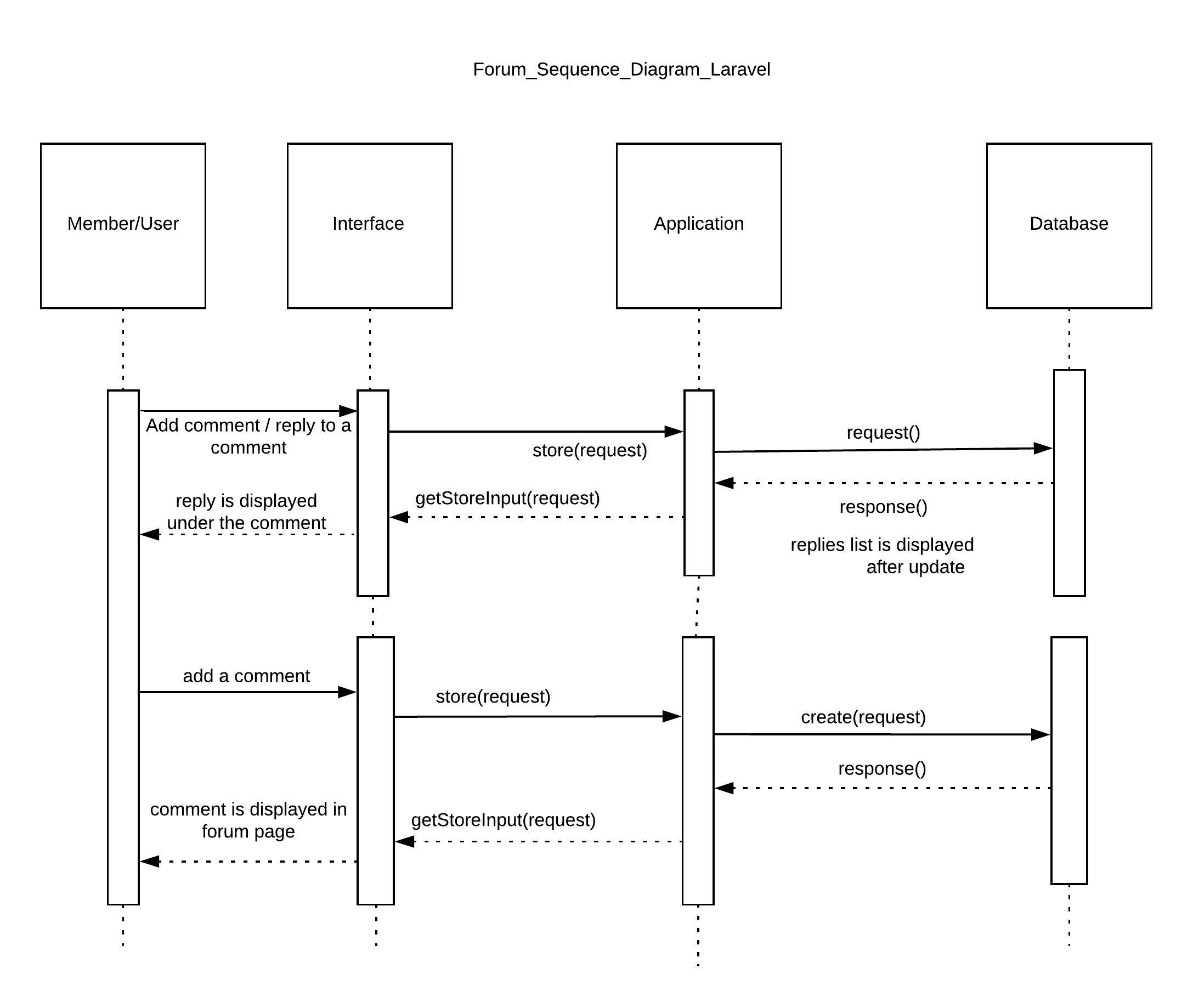
**Figure 18: User login sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.**

The user sends a request to show a form to application via interface and the application accepts the request and responds with the login form. The user fills up the login details in the login form and submits the form. The application validates the user details, if the validation is successful, then the application redirects to the dashboard page. If the validation is unsuccessful, the application returns the login page with validation errors.

****

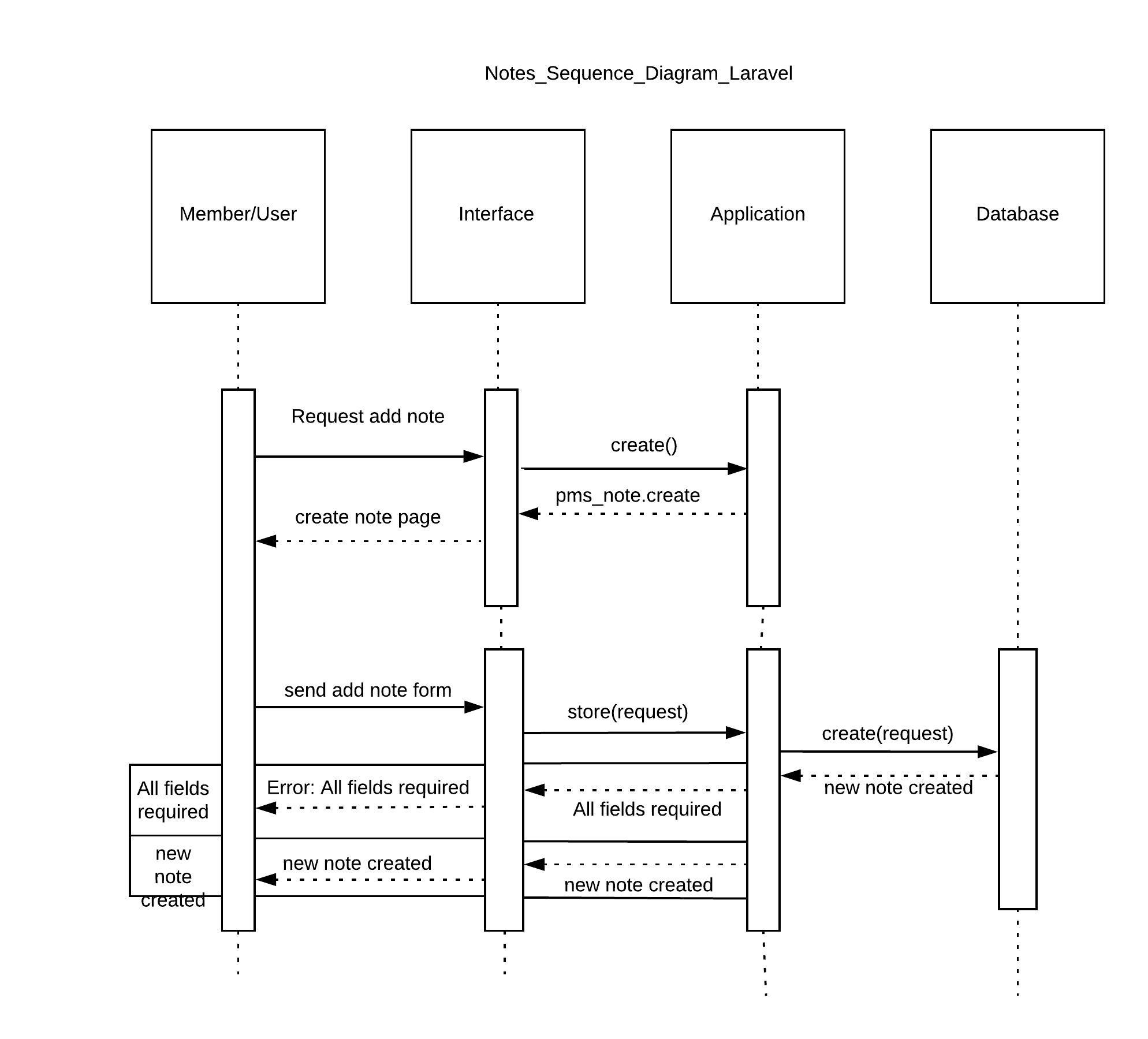
**Figure 19: Add/update documents sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.**

When the user clicks on the 'Add Document' button on the web application, the web browser (interface) sends an HTTP GET request to the web server (application). The web server accepts the HTTP GET request and the create() function is triggered in the document controller. Then it returns a view page which consists of a new document form markup code. It sends a response to the web browser containing the view and the web browser displays the new document form. The user fills up the fields in the new document form and submits the form to create a new document. The web browser sends an HTTP POST request to the web server (Application), the web server accepts the request and the store() function validates the required fields received from the posted data. If the validation is successful, the request is forwarded to create() function where the database query is generated based on the create request to the database server. The data is inserted into the database and notification of successful creation of a new document is sent to the web browser as response. If the validation is unsuccessful, the function returns the view page with the message that the required fields are not filled.

****

**Figure 20: Discussion forum sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.**

The user sends a request to application to show a forum page via the interface and the forum page is displayed. The user adds a new comment on the comment form and submits the form, which is then sent to the application and the application send the request to the database where the new comment gets inserted. The forum page is displayed with the list of the available comments. The user can reply to the comment posted, which is sent to the application and finally inserted into the database. The new replies list is displayed after the update.

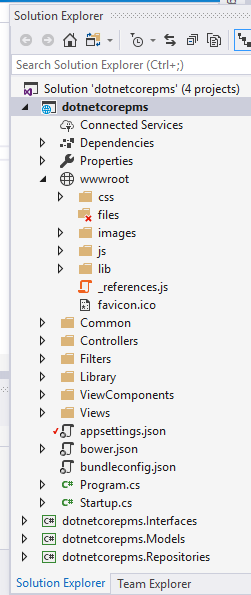
****

**Figure 21: Add/update notes sequence diagram of the Project Management Web Applications in Laravel on Windows and Ubuntu.**

The user sends a request to application to show a Notes page via the interface and the Notes page is displayed. The user adds a new Note and submits the note, which is then sent to the application and the application send the request to the database where the new note gets inserted along with the member’s name. The Note page is displayed with the list of the available notes. Also, the user can update the notes posted, which is sent to the application and finally inserted into the database.

**Appendix 3**

**Project Management File structure in .NET Core Framework**



**Figure 22: File Structure of the Project Management System in .NET Core in Visual Studio**

**Page Description of Project Management application in Laravel Framework**

Description of important pages that are added or modified for the building of the Project Management System web application in .NET Core Framework.

**Common/ModuleHelper.cs:** The ModuleHelper class is responsible for listing out the available menu items and helping the side bar navigation view to display the menu items as per the given order. It also helps to setup main menu items and their sub-menu items (if any) in the navigation sidebar.

**AdminController.cs:** The AdminController is responsible for displaying the dashboard of the application after the user is successfully logged in.

**BaseController.cs:** The BaseController, as the name suggests, is the base controller for all the controllers used in the application. The BaseController provides the basic and common functionalities to all the other controllers.

**DocumentsController.cs:** The DocumentsController is responsible to allow users to upload their document into the server and view the documents uploaded by other users in the application. The user can upload word document, text files via this feature.

**ErrorController.cs:** The ErrorController is responsible to redirect the application to an error page if any error occurs in the application.

**ForumController.cs:** The ForumController is responsible for providing a common platform for the users in the application to communicate with each other. It is a discussion forum where the users provide their comment on a particular topic and discuss on the topic with each other.

**NotesController.cs:** The NotesController is responsible for allowing the users to write their notes and share the notes among all the users in the PMS.

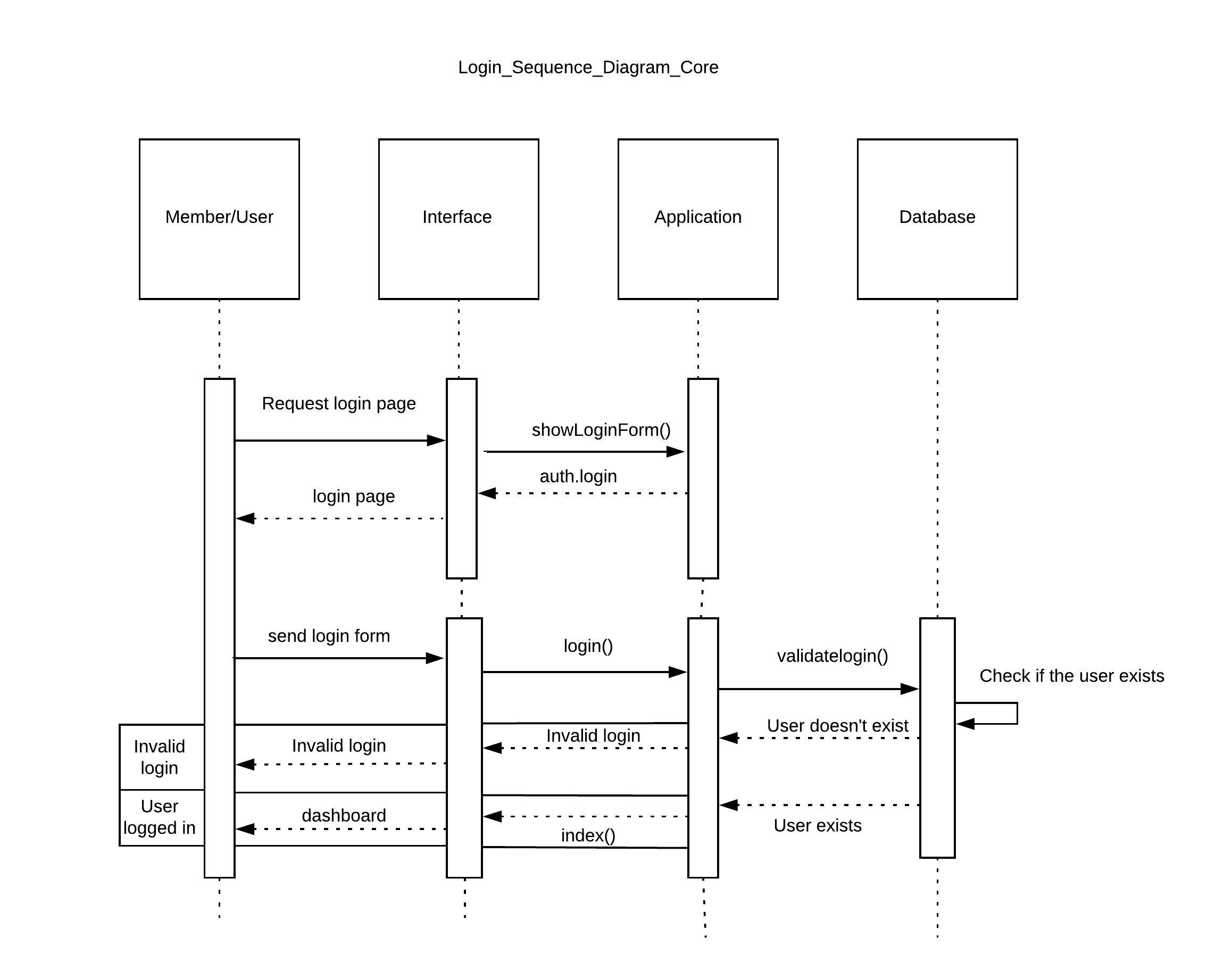
**RolesController.cs:** The RolesController is responsible for managing roles in the application. The roles such as administrator, normal user, etc. are created, deleted, viewed and updated under this feature in the PMS.

**UsersController.cs:** The UsersController is responsible for managing users in the application. The UsersController can be used to create, delete, view and update the users in the PMS.

**EncryptionLibrary.cs:** The EncryptionLibrary is responsible to encrypt and decrypt the password for authentication purposes

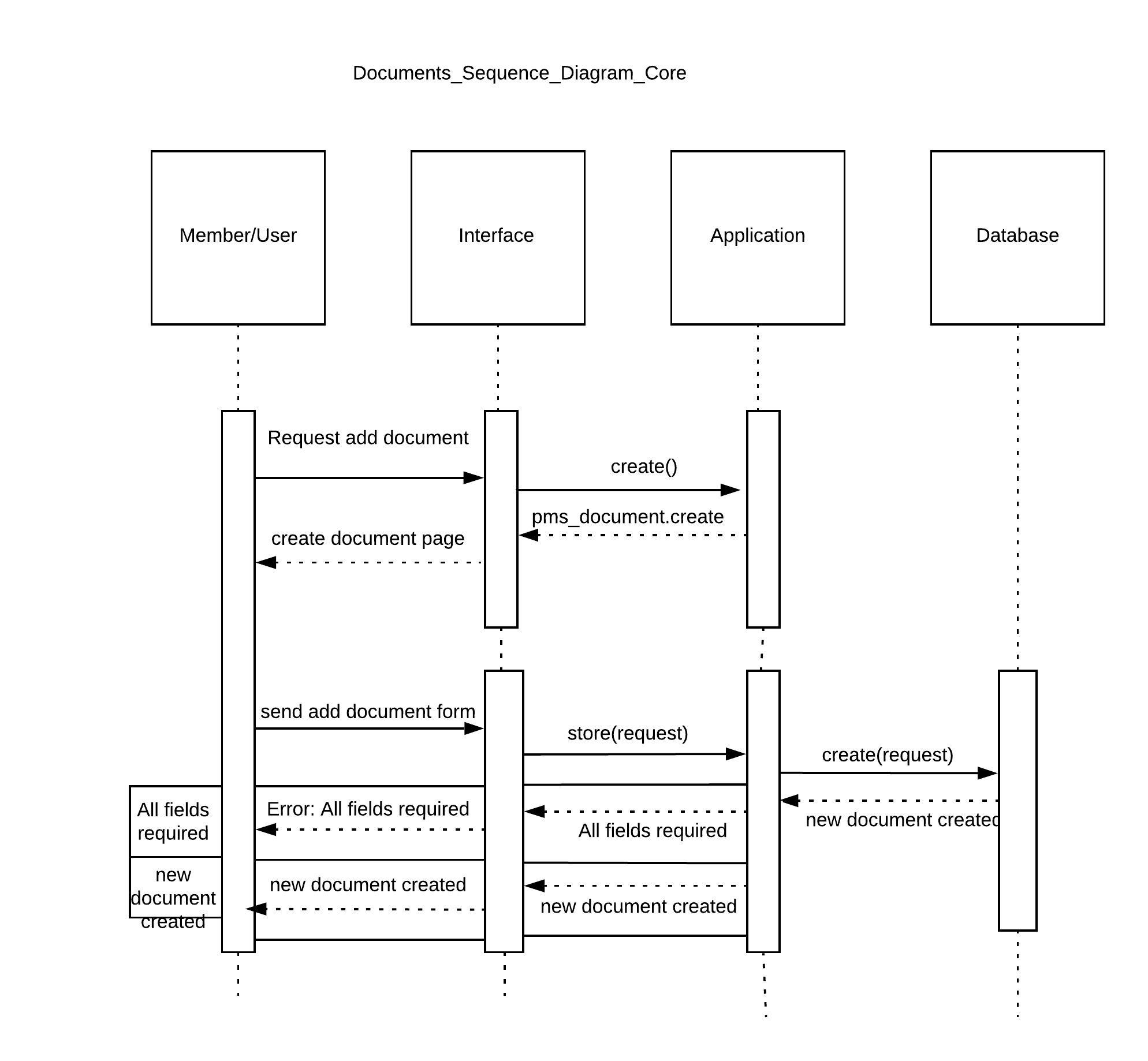
**Appendix 4**

**.NET Core Sequential Diagram**



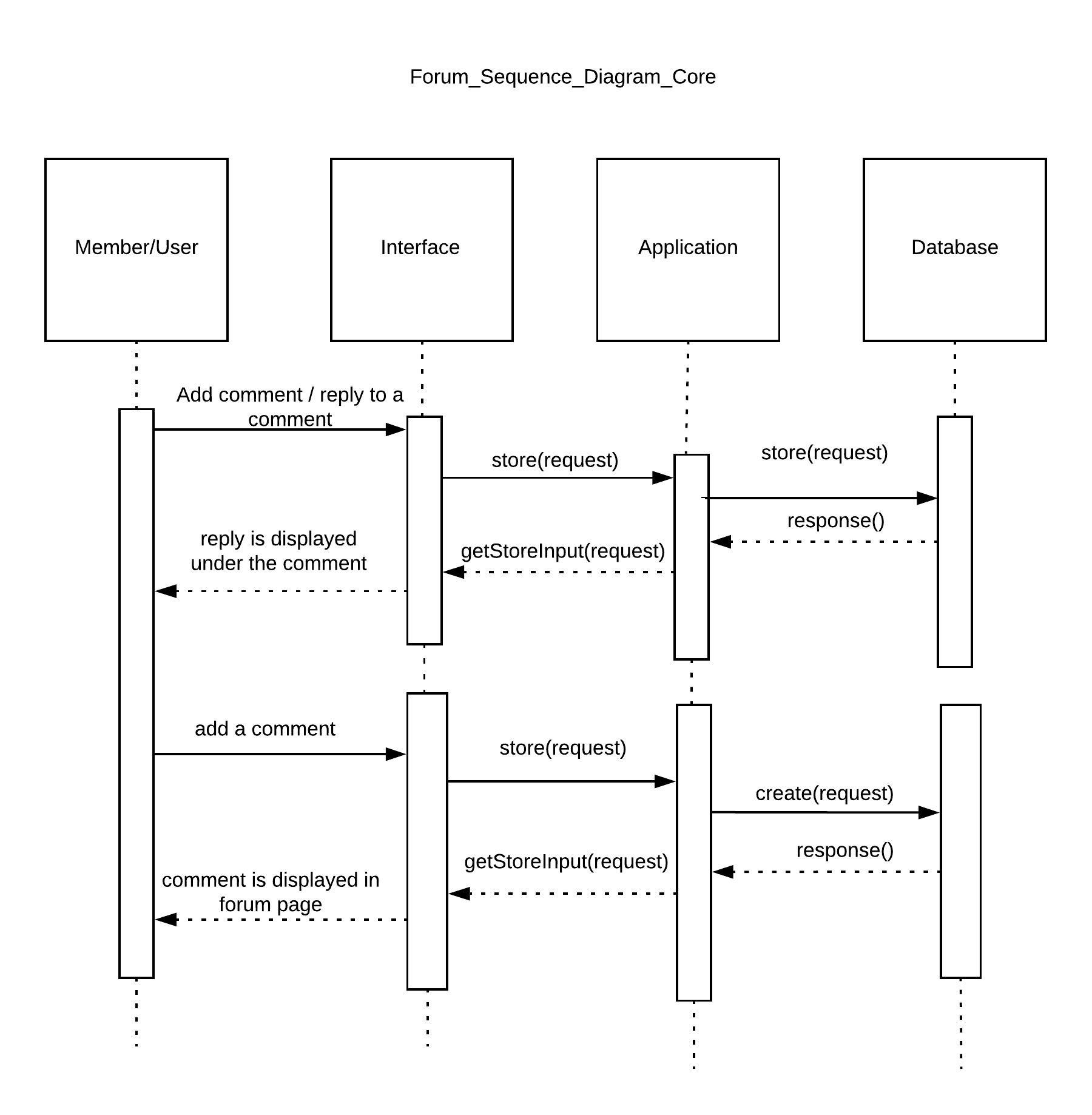
**Figure 22: User login sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.**

The user sends a request to show a form to application via interface and the application accepts the request and responds with the login form. The user fills up the login details in the login form and submits the form. The application validates the user details, if the validation is successful, then the application redirects to the dashboard page. If the validation is unsuccessful, the application returns the login page with validation errors.



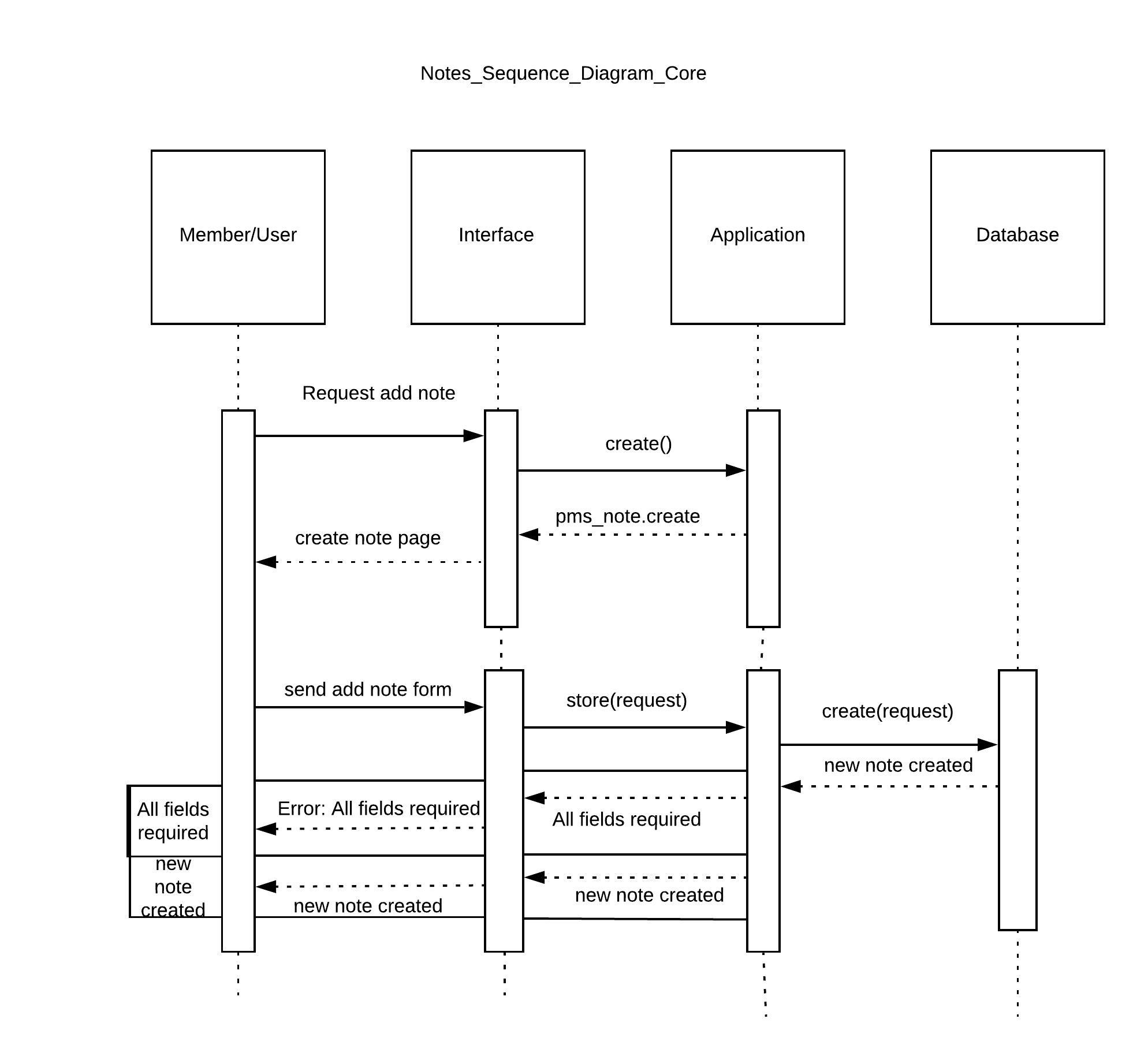
**Figure 23: Add/update documents sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.**

To add new document, the web browser (interface) sends an HTTP GET request to the web server (application). The document controller in the application gets triggered once the web server accepts the HTTP GET request from the web server. Then it returns a view page which consists of a new document form markup code. It sends a response to the web browser containing the view and the web browser displays the new document form. The user fills up the fields in the new document form and submits the form to create a new document. The web browser sends an HTTP POST request to the web server (Application), the web server accepts the request and the store() function validates the required fields received from the posted data. If the validation is successful, the request is forwarded to create() function where the database query is generated based on the create request to the database server. The data is inserted into the database and notification of successful creation of a new document is sent to the web browser as response. If the validation is unsuccessful, the function returns the view page with the message that the required fields are not filled.



**Figure 24: Discussion Forum sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.**

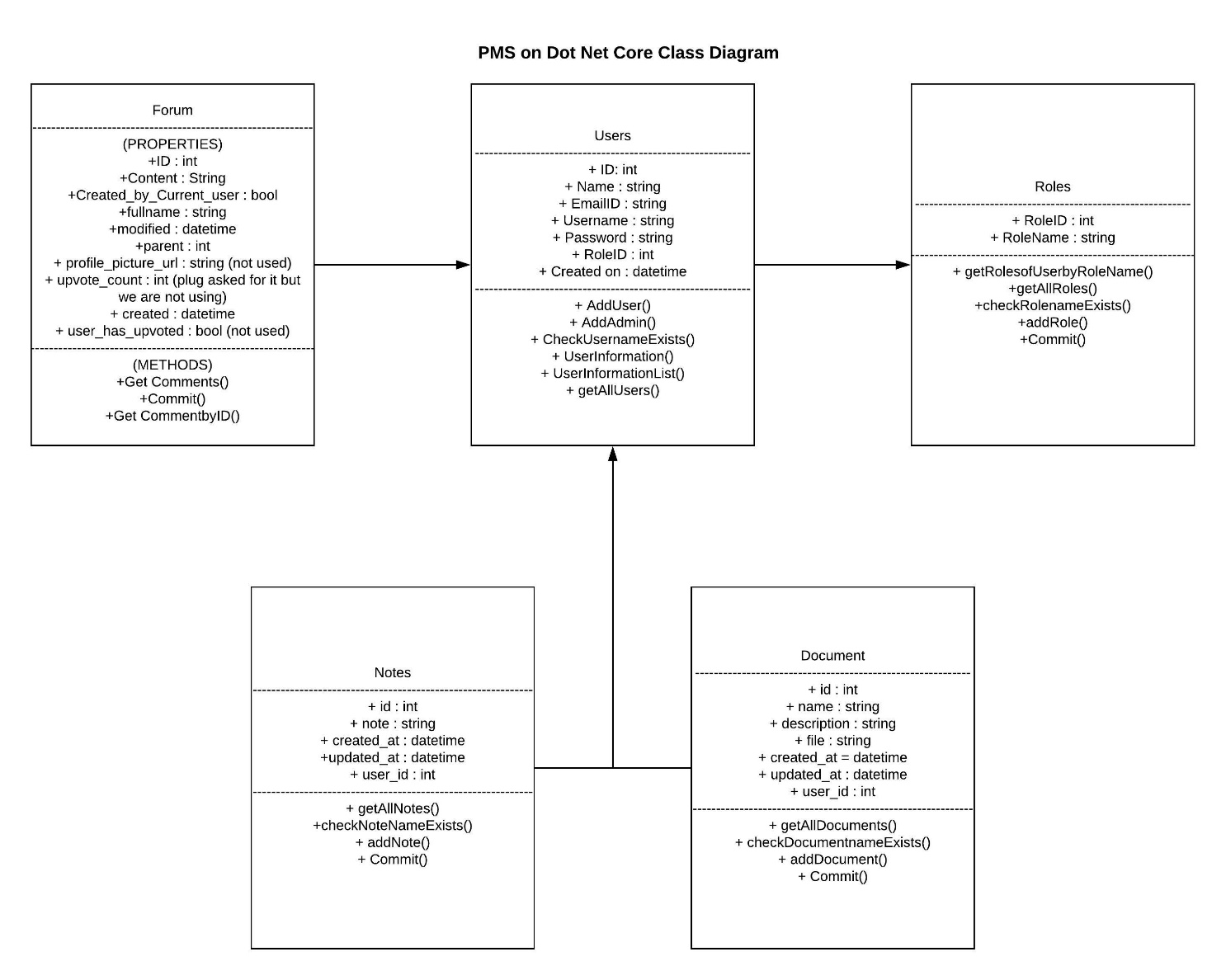
The user sends a request to application to show a forum page via the interface and the forum page is displayed. The user adds a new comment on the comment form and submits the form, which is then sent to the application and the application send the request to the database where the new comment gets inserted. The forum page is displayed with the list of the available comments. The user can reply to the comment posted, which is sent to the application and finally inserted into the database. The new replies list is displayed after the update.



**Figure 25: Add/Update Notes sequence diagram of the Project Management Web Applications in Dot Net Core on Windows.**

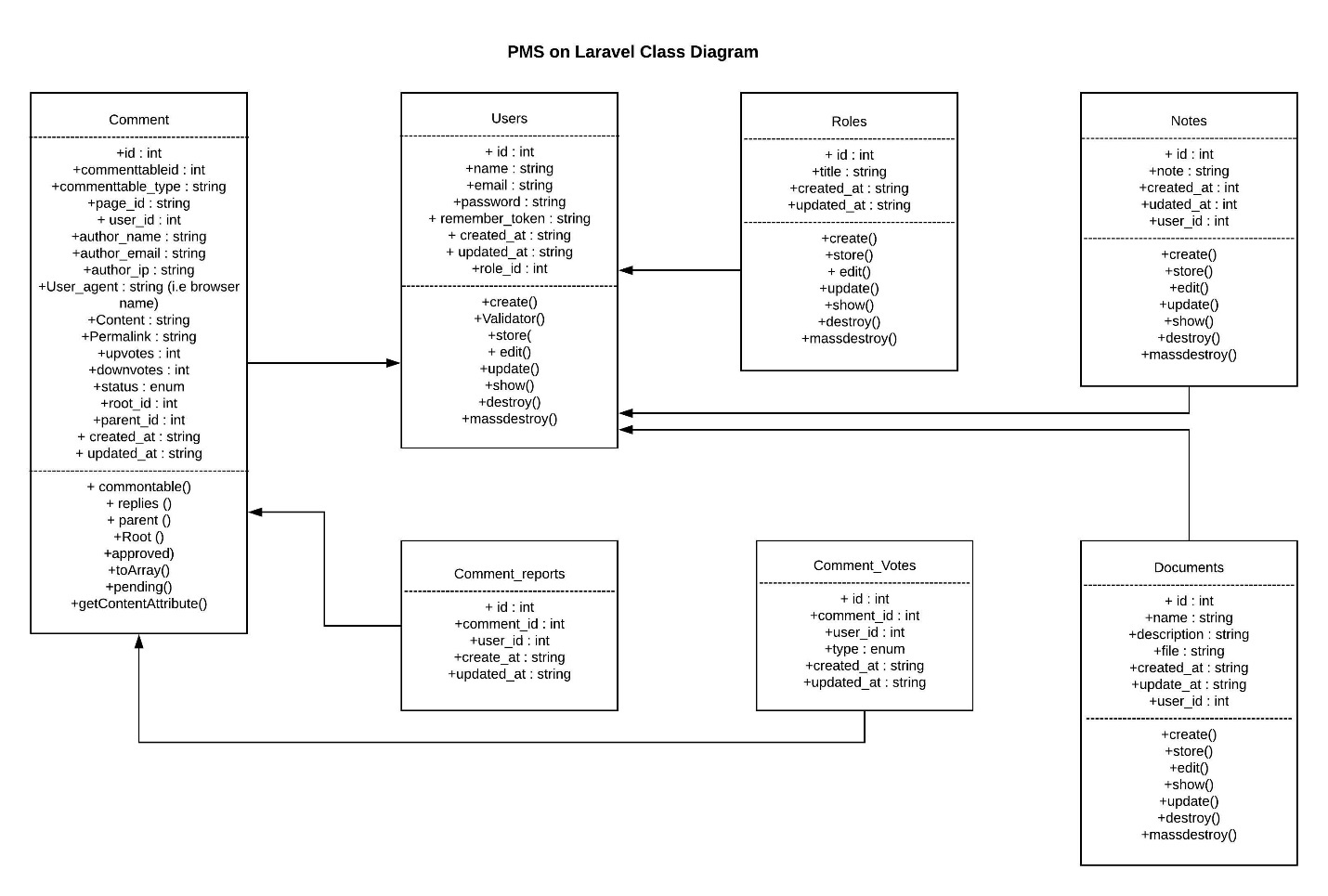
The user sends a request to application to show a Notes page via the interface and the Notes page is displayed. The user adds a new Note and submits the note, which is then sent to the application and the application send the request to the database where the new note gets inserted along with the member’s name. The Note page is displayed with the list of the available notes. Also, the user can update the notes posted, which is sent to the application and finally inserted into the database.

**Appendix 5**

****

**Figure 26: Class Diagram of the Project Management Web Application in Dot Net Core on Windows**

Figure 26 shows that the properties and methods of all classes are public. The Users class consists of properties such as Name, EmailID, Username, Password and is responsible to add, update, delete, validate, retrieve the user information. Then the Users class is accessed by Forum, Roles, Notes and Document classes to execute their methods and adds/updates/deletes/validate their associated properties.

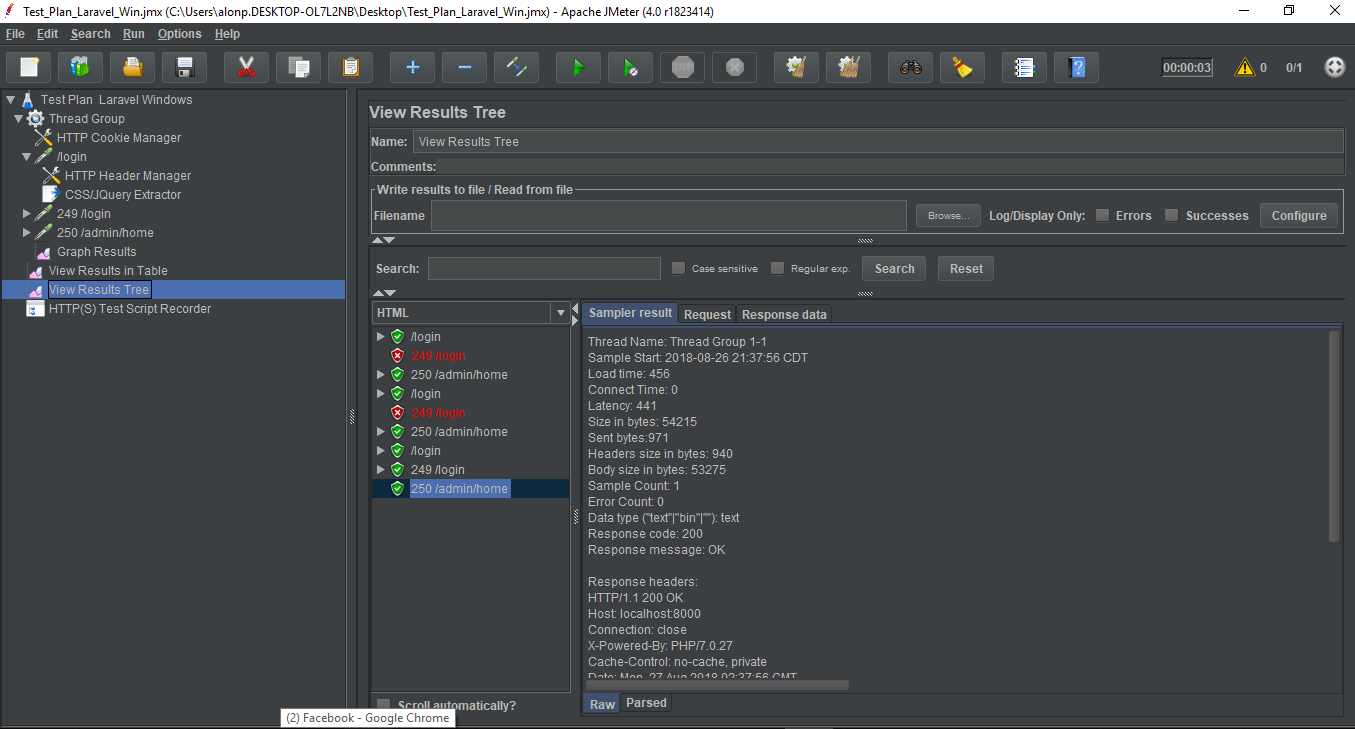
****

**Figure 27: Class Diagram of the Project Management Web Application in Laravel on Windows and Ubuntu**

Figure 27 shows that the properties and methods of all classes are public. The Users class consists of properties such as Name, EmailID, Username, Password and is responsible to add, update, delete, validate, retrieve the user information. Then the Users class is accessed by Forum, Roles, Notes and Document classes to execute their methods and adds/updates/deletes/validate their associated properties. Comment-reports and Comment\_Votes entities are classes which are responsible for associating comments with a user id and associate the comment votes on a particular comment of a particular user respectively. We are not using Comment-reports and Comment\_votes in this web application.

**Appendix 6**

**Testing of Project Management System**



**Figure 28: Testing of Project Management System with Jmeter**

Apache Jmeter is used to measure the performance of the Project Management System web application. The data on the page load time in all concerned environment was extracted through Jmeter. The pages that were considered during testing are /login, /admin/home, /admin/pms\_documents, and /admin/roles for web application in all three environments. These can be seen under the sub-heading, “Thread Group”. The results can be viewed in the sub-heading “View Results in Table” and “View Results Tree”. Since the Laravel application uses (Cross-site Reference Forgery) CSRF protection, it requires special and unique token to be validated every time. The JMeter cannot retrieve the pages which require authentication. The HTTP(S) Test Script Recorder creates a proxy server that acts as an intermediary between the browser and the application, retrieves the pages along the authentication page being accessed via the browser. Using CSS/JQuery Extractor, the retrieved authentication page is modified to mimic actual authentication to access the pages that require authentication. HTTP Cookie Manager is also added, since the authentication page requires cookie to store authentication details as session.

**References**

1. Bisson, S. (2017, March 15). .Net Framework or .Net Core? When to use which. Retrieved January 28, 2018, from <https://www.infoworld.com/article/3180478/development-tools/net-framework-or-net-core-when-to-use-which.html>
2. Building Web Applications From Scratch With Laravel. Retrieved September 15, 2017, from <https://code.tutsplus.com/tutorials/building-web-applications-from-scratch-with-laravel--net-25517>
3. Freeman, A. (2016). Pro ASP.NET Core MVC. Berkeley, CA: Apress.
4. Helm, R., Johnson, R. E., Gamma, E., & Vlissides, J. (2000). Design patterns elements of reusable object-oriented software. Charlesbourg, Québec: Braille Jymico Inc.
5. H. Komara, B. Hendradjay, and G. A. P. Saptawati, "Dynamic generic web pattern for multi-platform," *2016 International Conference on Data and Software Engineering (ICoDSE)*, Denpasar, 2016, pp. 1-5.  
   URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7936130&isnumber=7936094>
6. 1. L. (2017, November 21). Laravel/laravel. Retrieved November 21, 2017, from <https://github.com/laravel/larave>l
7. L. Alfat, A. Triwiyatno and R. R. 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)*, Semarang, 2015, pp. 46-51.  
   URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7437768&isnumber=7437747>
8. L. Chen, W. Huang, A. Sui, D. Chen and C. Sun, "The online education platform using Proxmox and noVNC technology based on Laravel framework," *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, Wuhan, 2017, pp. 487-491.  
   URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7960041&isnumber=7959951>
9. M. Jailia, A. Kumar, M. Agarwal and I. Sinha, "Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework," *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, 2016, pp. 1-5.  
   URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7892651&isnumber=7892628>
10. .NET Design Patterns. Retrieved May 21, 2017, from <http://www.dofactory.com/net/design-patterns>
11. Palleboina, S., & Eltaeib, T. (2015). A Study of MVC – A Software Design Pattern for Web: Application Development on J2ee Architecture. Journal of Multidisciplinary Engineering Science and Technology (JMEST), 2(4). Retrieved January 17, 2017
12. Should I learn and start doing ASP.NET Core? Retrieved January 28, 2018, from <https://codingblast.com/learn-asp-net-core/>
13. Stauffer, M. (2016). Laravel: up and running. S.l.: OReilly Media.
14. Understanding ASP.NET MVC. ASP.NET MVC 4 Recipes, 15-57.
15. X. Chi, B. Liu, Q. Niu and Q. Wu, "Web Load Balance and Cache Optimization Design Based Nginx under High-Concurrency Environment," *2012 Third International Conference on Digital Manufacturing & Automation*, GuiLin, 2012, pp. 1029-1032.  
    URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6298691&isnumber=6298247>
16. MVC Design Pattern. (2018, February 08). Retrieved October 9, 2018, from <https://www.geeksforgeeks.org/mvc-design-pattern/>