IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, July 2023

Agile Software Development with Laravel: Leveraging Object-Oriented Design and Framework Tools

Alma Christie C. Reyna

Faculty, College of Engineering and Information Technology,
Surigao Del Norte State University, Surigao City, Philippines

Abstract: *This paper highlights the synergy between Agile Software Development, Laravel Framework, and Object-Oriented Design (OOD) in creating flexible, efficient, and customer-centric software applications. Agile methodologies enable rapid adaptation to changing requirements, iterative development, and continuous feedback. Leveraging Laravel's features, task management, CI/CD integration, and OOD principles, developers can build high-quality applications that meet evolving customer needs. The seamless collaboration and streamlined development process ensure swift deployment and customer satisfaction, making this integration a powerful approach in the fast-paced digital landscape.*

Keywords: Laravel, Agile software development, object-oriented design

I. INTRODUCTION

Agile Software Development has emerged as a highly efficient and flexible approach to software project management, allowing teams to promptly adapt to changing requirements and deliver top-notch products with utmost efficiency. This methodology is deeply rooted in the Agile Manifesto, which prioritizes "individuals and interactions over processes and tools" and "responding to change over following a plan." Agile practices place a strong emphasis on customer collaboration, iterative development, and continuous feedback [1].In the Agile approach, software development is organized into manageable and incremental iterations referred to as sprints. Each sprint typically lasts a few weeks, during which cross-functional teams collaborate to deliver specific features or user stories. Throughout this process, regular feedback from stakeholders and end-users is collected to refine and prioritize the project's backlog, ensuring that the final product closely aligns with customer needs and expectations.The advantages of Agile Software Development are numerous, including heightened customer satisfaction, reduced time-to-market, improved team morale, and superior product quality due to continuous testing and validation. By fostering an environment of open communication and adaptability, Agile practices have become a fundamental cornerstone of contemporary software development.

Laravel, a PHP web application framework released as open-source software, has gained rapid popularity among developers due to its straightforwardness, expressive syntax, and robust capabilities [2]. Created by Taylor Otwell in 2011, Laravel adheres to the principles of elegant and efficient coding, rendering it an optimal choice for Agile software development. At the core of the Laravel framework lie essential components like the Eloquent ORM (Object-Relational Mapping) and Blade templating engine, which contribute to swift development and seamless integration with databases. With Eloquent, developers can efficiently interact with databases using Object-Oriented syntax, streamlining data manipulation and retrieval processes. Moreover, Laravel's modular architecture and extensive documentation foster efficient collaboration within Agile teams. The availability of a diverse range of pre-built packages and libraries within the Laravel ecosystem further aids in saving development time and reducing effort.

Object-Oriented Design (OOD) is a programming paradigm that empowers developers to represent real-world entities as self-contained "objects," encompassing both data and behavior. This design philosophy fosters the reuse, maintainability, and scalability of code, aligning seamlessly with Agile principles, which emphasize adaptability and iterative development. The application of design patterns and solid principles in OOD further enhances code quality and reduces technical debt [3]. By organizing code into interconnected and modular units, OOD facilitates smooth

DOI: 10.48175/IJARSCT-12165

collaboration among Agile team members and simplifies the process of introducing changes or incorporating new features [4][11].

This study aims to delve into the effective synergy between Agile Software Development, the Laravel Framework, and Object-Oriented Design. By harnessing these complementary methodologies, developers can craft software that is flexible, efficient, and of high quality, effectively meeting the ever-changing demands of today's fast-paced digital landscape.

II. LEVERAGING LARAVEL FRAMEWORK TOOLS

2.1 Key Features of Laravel for Agile Development

Laravel, a potent PHP web application framework, presents a wide array of features that harmonize effectively with the principles of Agile software development. These crucial attributes empower development teams to construct efficient, scalable, and customer-centric applications while maintaining a swift development pace.

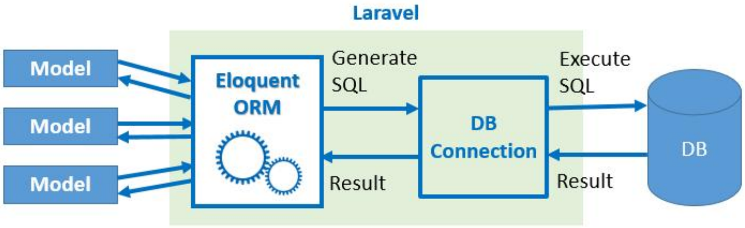
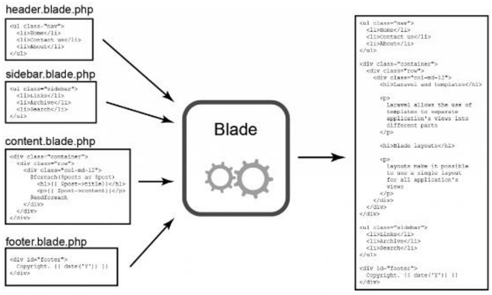


Fig. 1. Laravel Eloquent ORM

Laravel's Eloquent ORM employs an expressive syntax that enables developers to interact with databases using an intuitive, Object-Oriented approach, streamlining data manipulation and retrieval [2]. Fig. 1 shows the Laravel EloquentORM that uses Active Record Pattern which is a technique to wrap database into objects. By using eloquent, we can present data stored in a database table as a class and records as objects. So each database table has "Model" which is used to interact with the table. We can use eloquent to create, edit, query and delete records easily rather than raw SQL. It makes the code easily readable and much cleaner. This feature enhances developers' productivity and facilitates code maintainability during Agile iterations, allowing for rapid adaptation to changing requirements.



```
<ul class="nav">
  <li>Home</li>
  <li>Contact us</li>
  <li>About</li>
</ul>
```

```
<ul class="sidebar">
  <li>Links</li>
  <li>Archive</li>
  <li>Search</li>
</ul>
```

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      @foreach($posts as $post)
        <h1>{{ $post->title}}</h1>
        <p>{{ $post->content}}</p>
      @endforeach
    </div>
  </div>
</div>
```

```
<div id="footer">
  Copyright. {{ date('Y')} }}
</div>
```

```
<ul class="nav">
  <li>Home</li>
  <li>Contact us</li>
  <li>About</li>
</ul>

<div class="container">
  <div class="row">
    <div class="col-md-12">
      <h1>Laravel and templates</h1>

      <p>
        Laravel allows the use of
        templates to separate
        application's views into
        different parts
      </p>

      <h1>Blade layouts</h1>

      <p>
        Layouts make it possible
        to use a single layout
        for all application's
        views
      </p>
    </div>
  </div>
</div>

<ul class="sidebar">
  <li>Links</li>
  <li>Archive</li>
  <li>Search</li>
</ul>

<div id="footer">
  Copyright. {{ date('Y')} }}
</div>
```

Fig. 2. Laravel blade templating

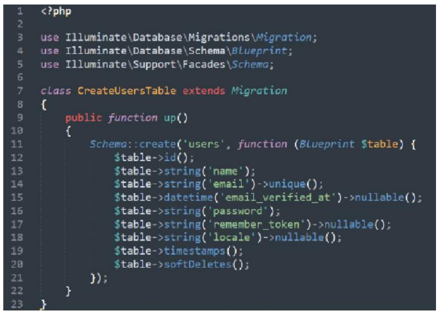
Furthermore, Laravel incorporates built-in support for the blade templating engine (Fig. 2), ensuring a clear and maintainable separation between frontend and backend code. This characteristic facilitates Agile teams to collaborate efficiently on different sections of the application simultaneously, promoting streamlined cooperation and expediting development progress.

2.2 Database Migrations and Seeders for Rapid Development

Database Migrations and Seeders in Laravel play a crucial role in accelerating the development process and ensuring consistency among team members. Migrations facilitate database schema changes in a version-controlled manner, allowing developers to modify the database structure without manual intervention [5].

DOI: 10.48175/IJARSCT-12165

ISSN 2581-9429 IJARSCT



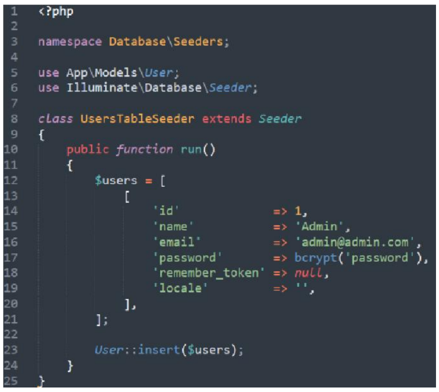
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->datetime('email_verified_at')->nullable();
            $table->string('password');
            $table->string('remember_token')->nullable();
            $table->string('locale')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });
    }
}
```

Fig. 3. Table migration

During Agile iterations, the database schema may evolve rapidly due to changes in requirements. With Migrations, developers, can easily apply schema changes across different environments, ensuring a smooth and reliable deployment process.Fig. 4 shows a sample table migration.



```php
<?php

namespace Database\Seeders;

use App\Models\User;
use Illuminate\Database\Seeder;

class UsersTableSeeder extends Seeder
{
    public function run()
    {
        $users = [
            [
                'id'             => 1,
                'name'           => 'Admin',
                'email'          => 'admin@admin.com',
                'password'       => bcrypt('password'),
                'remember_token' => null,
                'locale'         => '',
            ],
        ];

        User::insert($users);
    }
}
```

Fig. 4. Table seeder

Seeders (Fig. 5), on the other hand, provide a convenient way to populate the database with test or sample data. Agile teams can utilize Seeders to create a consistent and reliable testing environment, enabling efficient testing and validation during each sprint.

2.3 Authentication, Authorization, and Testing Tools in Laravel

Laravel streamlines the critical processes of authentication and authorization in web applications through its integrated authentication system [6]. This simplification empowers Agile teams to swiftly incorporate user authentication functionalities like registration, login, and password reset, enabling them to concentrate on delivering essential application features.

Moreover, Laravel's authorization capabilities offer precise control over user permissions and access to specific sections of the application [7]. Agile teams can dynamically adjust user access levels throughout the development journey, accommodating evolving project requirements.

To ensure code quality and application reliability, Laravel provides robust testing tools, including seamless integration with PHPUnit [9]. Agile teams can seamlessly incorporate Test-Driven Development (TDD) practices into their workflow, ensuring continuous validation and improvement of the application.

By leveraging Laravel's authentication, authorization, and testing tools, Agile teams can accelerate the development process while maintaining the security and excellence of the final product.

III. APPLYING OBJECT-ORIENTED DESIGN IN LARAVEL

3.1 Fundamentals of Object-Oriented Design (OOD) Principles

Object-Oriented Design (OOD) is a programming approach that provides developers with the ability to conceptualize software systems as a set of interacting objects, where each object encapsulates both data and behaviour [10]. A profound grasp of the fundamental principles of OOD holds utmost importance in constructing well-organized and easily maintainable Laravel applications.

DOI: 10.48175/IJARSCT-12165

Within the realm of OOD, developers abide by key principles, namely Encapsulation, Inheritance, Polymorphism, and Abstraction (Encyclopedia of Software Engineering). These principles play a pivotal role in organizing code and elevating the application's adaptability and scalability, perfectly aligning with the principles of Agile software development.

3.2 Implementing OOD in Laravel

Applying Object-Oriented Design principles in Laravel can greatly enhance the structure and clarity of code within controllers and models. By embracing the OOD principle of Encapsulation, developers have the capability to group related methods and properties within classes, leading to better code organization and minimizing the risk of conflicts [13][14].

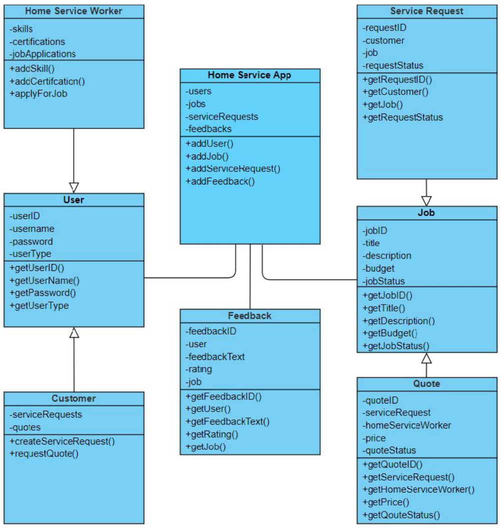


Fig. 5. Class diagram

Fig. 3 shows a class diagram constructed during object-oriented design stage can be used as the template in implementing the application in Laravel controllers and models [9][17]. Moreover, making use of the principle of inheritance empowers developers to establish subclasses that inherit properties and behaviors from their parent classes. This strategy fosters code reusability and maintains uniformity throughout the entire codebase of the application.

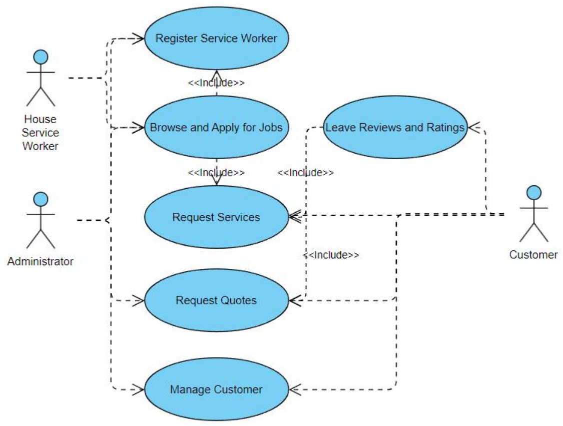


Fig. 6. Use-case diagram

On the other hand, the use-case diagram, a sample is shown in Fig. 7, created during the object-oriented design stage is essential in the development of web applications using the Laravel framework. The use-case diagram serves as a valuable tool for understanding the system's functionalities from a user's perspective, identifying the different actions users can perform and their interactions with the application [12]. Also, Laravel blade templates play a crucial role in implementing the presentation layer of the Model-View-Controller (MVC) architecture in Laravel. They are used to

DOI: 10.48175/IJARSCT-12165

define the user interface and handle the rendering of dynamic content based on the use-cases identified in the diagram. By aligning the functionalities depicted in the use-case diagram with the actual implementation in Laravel Blade templates, developers can ensure that the web application's views effectively fulfill the specified use-cases, resulting in a user-friendly and intuitive interface for the application's users. This integration of the use-case diagram and Laravel Blade templates enhances the overall usability and user experience, contributing to the success of the Laravel-based application.

Leveraging Interfaces and Traits for code reusability

In Laravel, Interfaces and Traits present potent mechanisms for achieving code reusability, a fundamental aspect of Agile software development. Interfaces establish a contract for classes that implement them, ensuring the implementation of specific methods in those classes [14]. By adhering to the Interface segregation principle, developers can attain loose coupling and foster more adaptable code maintenance.

On the other hand, Traits offer developers the ability to include methods in classes without forming a full inheritance relationship (Laravel Documentation). This trait-based approach facilitates the reuse of common functionalities across multiple classes, promoting modular code and diminishing redundancy.

By incorporating Object-Oriented Design principles in Laravel controllers and models, and effectively leveraging interfaces and traits for code reusability, developers can construct resilient, flexible, and easily maintainable applications that align seamlessly with the Agile principles of adaptability and iterative development.

IV. INTEGRATING AGILE PRACTICES WITH LARAVEL

4.1 Managing Agile projects with Laravel tools

In Agile development, Laravel offers a comprehensive set of tools and functionalities that support efficient project management. By utilizing Laravel's inherent task management capabilities and version control systems, Agile teams can seamlessly track and arrange tasks within designated sprints. Moreover, Laravel's integrated user authentication and authorization features ensure secure and controlled access to project resources, promoting effective collaboration among team members throughout the entire development lifecycle [14].

4.2 Continuous Integration and Deployment (CI/CD) in Laravel

In the context of Agile development, Continuous Integration and Deployment (CI/CD) hold significant importance, as they facilitate the swift and automated delivery of software updates. Laravel seamlessly integrates with CI/CD pipelines, empowering developers to automate essential tasks like testing, building, and deployment. By combining Laravel with widely-used CI/CD tools such as Jenkins, Travis CI, or GitHub Actions, Agile teams can achieve seamless integration and deployment, resulting in rapid feedback cycles and reduced time-to-market [15].

4.3 Sprint planning, backlog management, and collaboration with Laravel in Agile teams

Laravel's flexible ecosystem fully supports Agile methodologies in sprint planning and backlog management. Agile teams can take advantage of Laravel's task prioritization functionalities, along with the adaptability of its database migrations, to easily adjust and modify the project backlog during sprint planning sessions. Moreover, Laravel's integrated communication tools, including real-time notifications and event broadcasting, facilitate smooth collaboration and communication among Agile team members, enabling quick responses to changes and ensuring successful sprint outcomes [16].

By utilizing Laravel's project management tools, incorporating CI/CD pipelines, and making the most of its collaboration features, Agile teams can effectively align Laravel development with the core principles of Agile. This harmonious integration empowers teams to pursue iterative and customer-centric development, resulting in the timely and efficient delivery of high-quality software products.

V. CONCLUSION

In conclusion, the combination of Agile Software Development with Laravel and Object-Oriented Design empowers software developers to create flexible and high-quality applications. By leveraging Laravel's extensive features,

DOI: 10.48175/IJARSCT-12165

seamless task management, and CI/CD integration, developers can efficiently deliver customer-focused solutions, adapting to changing requirements and ensuring swift deployment. The synergy between Agile methodologies, Laravel, and OOD results in a dynamic development process that effectively meets the demands of modern software development.

REFERENCES

[1]. Agile Manifesto. Agile Alliance. https://agilemanifesto.org/

[2]. Laravel. "A PHP Framework For Web Artisans." Laravel. https://laravel.com/

[3]. Gamma, Erich, et al. "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley Professional, 1994.

[4]. Martin, Robert C. "Clean Architecture: A Craftsman's Guide to Software Structure and Design." Prentice Hall, 2017.

[5]. Laravel Migrations. Laravel Documentation. https://laravel.com/docs/8.x/migrations

[6]. Laravel Authentication. Laravel Documentation. https://laravel.com/docs/8.x/authentication

[7]. Laravel Authorization. Laravel Documentation. https://laravel.com/docs/8.x/authorization

[8]. Laravel Testing. Laravel Documentation. https://laravel.com/docs/8.x/testing

[9]. Havelund, K., Tegeler, T., Smyth, S., & Steffen, B. (2022, October). Discussing the Future Role of Documentation in the Context of Modern Software Engineering (ISoLA 2022 Track Introduction). In International Symposium on Leveraging Applications of Formal Methods (pp. 3-9). Cham: Springer Nature Switzerland.

[10]. Encyclopedia of Software Engineering. (2002). Object-Oriented Design Principles. John Wiley & Sons, Inc.

[11]. Griffin, J., & Griffin, J. (2021). Modularizing Laravel. Domain-Driven Laravel: Learn to Implement Domain-Driven Design Using Laravel, 237-284.

[12]. Gallera, J. (2023) , "Design and Evaluation of an Online Beach House Rental System: Streamlining Accommodation Management and Enhancing User Experience, Volume 10 - Issue 5, May 2023 Edition, p1421-1427.

[13]. Mangca, D. (2023). Assessment of a Web-Based Boarding House Booking System with Notification Capability. International Research Journal of Advanced Engineering and Science, 8(2), 59-62.

[14]. Laravel Documentation. (n.d.). Laravel: The PHP Framework for Web Artisans. Retrieved from https://laravel.com/docs/8.x

[15]. Laravel CI/CD. (n.d.). Laravel News. Retrieved from https://laravel-news.com/category/ci-cd

[16]. Laravel Communication. (n.d.). Laravel Echo. Retrieved from https://laravel.com/docs/8.x/broadcasting

[17]. Raglianti, M., Nagy, C., Minelli, R., Lin, B., & Lanza, M. (2023). On the Rise of Modern Software Documentation (Pearl/Brave New Idea). In 37th European Conference on Object-Oriented Programming (ECOOP 2023). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

DOI: 10.48175/IJARSCT-12165