Fall 2022 CSCI 445 Exam – Lauren Sherburne

**List 5 reasons to use a framework**

1. Increases Software Reliability – Since frameworks provide standard, low-level functionality, developers can "focus [their] efforts on the elements that makes the project unique" (Ranjan).
2. Save time – Using a framework speeds up programming time because it allows a programmer to implement a feature or design using a pre-built system. For example, CodeIgniter is a lightweight framework that saves time because less code is needed to accomplish a task (Ranjan).
3. Simplifies Testing – Developing using a framework provides ease of debugging, testing, and maintenance "because there is a community of developments associated with each framework" (Murtaza).
4. Improve Security – Frameworks typically have built-in security features and methods that developers can use so that they have an easier time securing their systems. (Murtaza)
5. Flexibility – Most frameworks are very flexible and can be used as a starting point in a project in order to build something much more complex (Naskar).

**Explain the MVC architecture. List (but don't describe) two alternatives.**

The Model-View-Controller, or MVC, framework utilizes a design pattern that "separates the application into three main logical components": the Model, the View, and the Controller (GeeksforGeeks). In a large and complex application system, using MVC simplifies development because each section of the application is given a specific purpose instead of multiple overlapping jobs. As shown in Figure 1 below, the model component handles all of the data related logic that the user works with and the view component handles all of the user interface. The views and model are connected because the views are created using data from the model. However, the view and the model only interact with the controller, not with one another. Thus, the controller enables a connection between the view and the model by processing all business logic and incoming requests. It is what produces the final output.

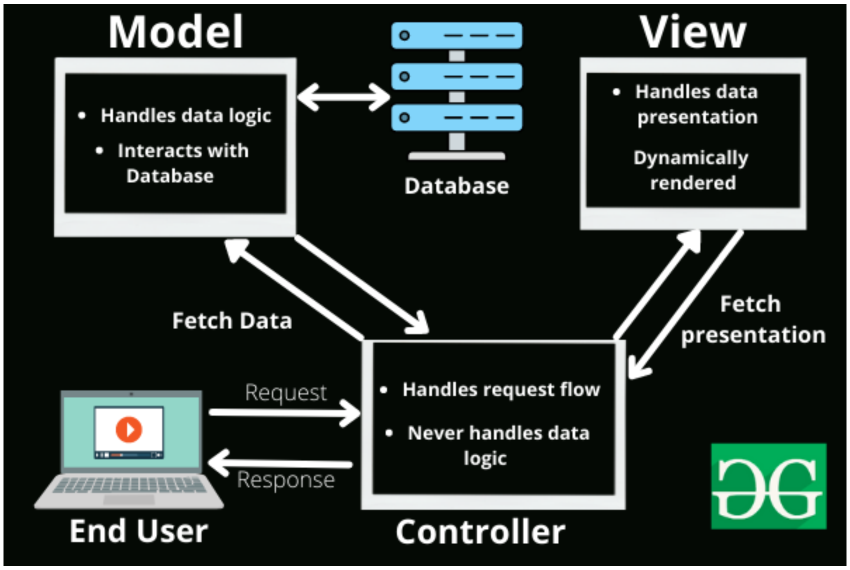


*Figure 1: MVC Framework Model (GeeksforGeeks)*

Two alternatives to the MVC framework include (1) the Model-View-ViewModel and (2) the Heirarchical Model-View-Controller.

**What is meant by a "lightweight" framework? When might this be your best option?**

The term "lightweight" when describing a framework refers to how much code is required to implement the framework. A lightweight framework "does not require as many code changes to incorporate them into your application as does a more heavyweight framework" ("What Is Mean by a

Lightweight Framework?”). One example of a lightweight framework is Spring since it allows minimally invasive implementation which saves time during development (Stalin).

Using a lightweight framework may be your best option if your team is new to using frameworks and you need to consider how the learning curve is going to affect your development process (“How to Choose Between Lightweight or Heavyweight Framework for Web Application.”). The lightweight framework, since it requires less code, could be easier to implement.

**Explain (in some detail) at least 2 security features commonly built into web frameworks**

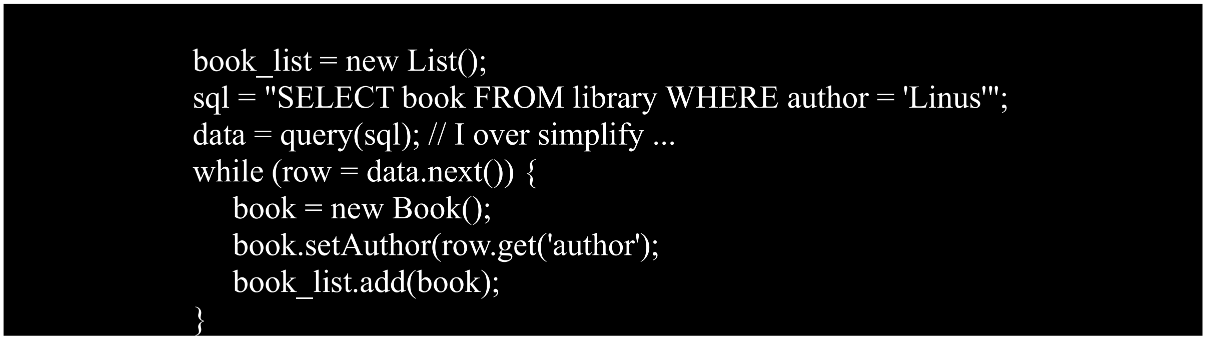
1. Many web frameworks have specific security features built to ensure database security. The frameworks parameterize database queries in order to block SQL injection attacks and they secure the database against cross-site scripting attacks by using templates (Boyer). It is important to note that these techniques are not 100% secure, but they take care of most XSS and SQLi vulnerabilities (Boyer). The following web frameworks protect against both SQL injection attacks and cross-site scripting attacks: Django, Ruby on Rails, and ASP.NET (Boyer).
2. Another common security feature built into web frameworks is the authentication and authorization of users. This allows the framework to restrict accesses based on user roles and attributes, and it involves implementing a username/password system for the web application (Boyer). Both Django and ASP.NET are web frameworks that authenticate users (Boyer).

**Explain Object-Relational Mapping (ORM)**

Developers using object-oriented programming need a way to utilize their database structure within their programming code. Thus, object-relational mapping is “ a technique that creates a layer between the language and the database, helping programmers work with data without the OOP paradigm” (Lang). Developers can access the database without having to know how the database is structured or how to write SQL. Instead, they use object-relational mapping which shows developers the relationship between a database table and a programming object (Lang). In other words, high-level abstraction simplifies the process of accessing the database.

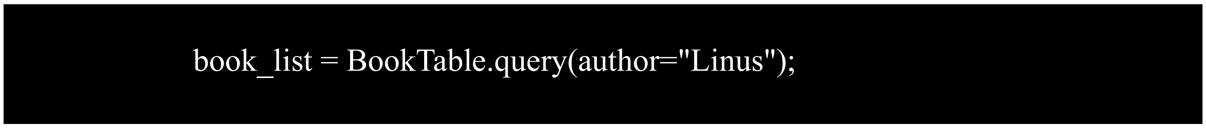
Typically, ORM is accomplished using a library so that the developer can continue using their language of choice to interact with objects. For example, if one of your database classes/tables is “book” and you want to query the database to find “all the books of which the author is ‘Linus’ ”, the following example shows how you would accomplish this using a direct SQL query versus an ORM library (“What Is an ORM, How Does It Work, and How Should I Use One?”):

Manual SQL Query:



```
book_list = new List();
sql = "SELECT book FROM library WHERE author = 'Linus'";
data = query(sql); // I over simplify ...
while (row = data.next()) {
    book = new Book();
    book.setAuthor(row.get('author');
    book_list.add(book);
}
```

ORM Library:



```
book_list = BookTable.query(author="Linus");
```

Examples citation: “What Is an ORM, How Does It Work, and How Should I Use One?”

**Explain (in some detail) the concept of routes and controllers.**

In a framework, the route and controller work together to implement navigation for an application. A router typically creates a list of all the routes in the application and it associates each route with a controller file that contains the implementation code (Coder). In web development specifically, a route is a section of “code that associates an HTTP verb (GET, POST, PUT, DELETE, etc.), a URL path/pattern, and a function that is called to handle that pattern” (MDN Contributers). As seen in Figure 2 below, the route forwards supported requests and the necessary information to the appropriate controller functions, and the controller functions get the requested data from the models before returning the proper HTML page populated with data to the user to view in the browser (MDN Contributers).

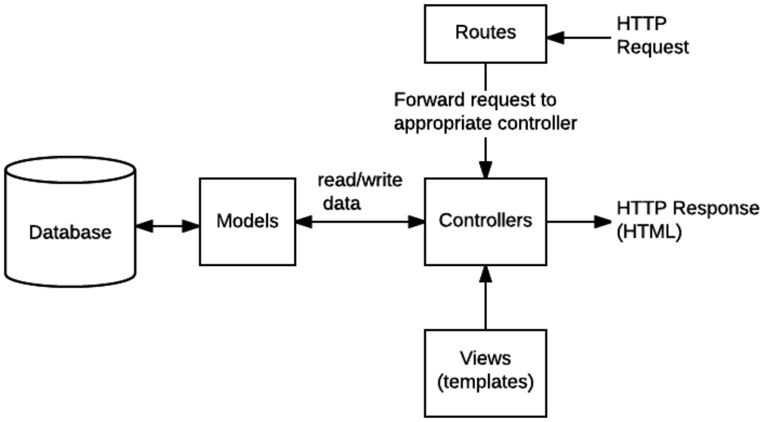


*Figure 2: Framework Routes and Controllers (MDN Contributers)*

When using Laravel, a popular web application framework, a basic controller used to show the profile for a given user looks like:

```
<?php
namespace App\Http\Controllers;
use App\Models\User;
class UserController extends Controller
{
   /**
    * Show the profile for a given user.
    *
    * @param  int  $id
    * @return \Illuminate\View\View
    */
   public function show($id)
   {
      return view('user.profile', [
         'user' => User::findOrFail($id)
      ]);
   }
}
```

and the corresponding route for the above controller method looks like:

```
use App\Http\Controllers\UserController;

Route::get('/user/{id}', [UserController::class, 'show']);
```

Examples citation: “The PHP Framework for Web Artisans.”

**Explain the REST architecture. What makes an API restful?**

The REST architecture stands for Representational State Transfer and it is a way to simplify the communication between computer systems on the web. A restful API sytem is characterized by how it is “stateless and separate[s] the concerns of client and server” (Codecademy Team). The separation of client and server simply means that the implementation of the client and server are completely independent of one another. In other words, the client-side code can be changed at any time without affecting the server-side code and vice versa (Codecademy Team). This separation improves the web application’s flexibility and scalability since each side can change and evolve separately. It is important to note that since the client and server need to communicate with one another, each side must know how to format the messages that are sent to the opposite side.

The stateless characteristic of a restful API system means that the architecture uses nouns, or resources, instead of verbs, or commands which means the system does not rely on the implementation of interfaces. A resource is used to “describe any object, document, or thing that you may need to store or send other services” (Codecademy Team). Therefore, the rest architecture allows the server to operate without knowing the client’s state and the client operates without knowing the server’s state.

A restful API includes the following constraints:

1. The client and server act independently.
2. The server does not record the state of the client.
3. The server marks whether data is cacheable.
4. The client and server interact in a uniform and predictable way (server exposes resources).
5. The application behaves the same regardless of any intermediaries between the client and server.

**What is CRUD, and how does this relate to REST?**

CRUD stands for create, read, update, and delete, and it is an acronym used to describe the major functions or data manipulations that are needed to successfully interact with a database. A REST application, in the same way, is built around resources that can be created, read, updated, and deleted (Bush). The REST API, however, uses HTTP’s GET, POST, PUT, DELETE, etc. to manipulate data just like a database’s CRUD functions. Therefore, the CRUD functions can in a way be mapped to the REST API’s functions (Bush):

- Create: POST, PUT
- Read: GET
- Update: POST, PUT, PATCH
- Delete: DELETE

# Works Cited

Boyer, Justin. “How Secure Are Popular Web Frameworks? Here Is a Comparison.” *Veracode*, www.veracode.com/blog/secure-development/how-secure-are-popular-web-frameworks-here-comparison. Accessed 17 Oct. 2022.

Bush, Thomas. “Crud vs. Rest: What's the Difference?” Nordic APIs, 31 July 2020, https://nordicapis.com/crud-vs-rest-whats-the-difference/.

Codecademy Team. “What Is Rest?” Codecademy, https://www.codecademy.com/article/what-is-rest.

Coder, Eric The. “NodeJS + Express Part 5: Routes and Controllers.” DEV Community, 1 Nov. 2021, dev.to/ericchapman/nodejs-express-part-5-routes-and-controllers-55d3.

GeeksforGeeks. “MVC Framework Introduction.” GeeksforGeeks, 13 Oct. 2022, www.geeksforgeeks.org/mvc-framework-introduction.

“How to Choose Between Lightweight or Heavyweight Framework for Web Application.” *Software Engineering Stack Exchange*, 25 Feb. 2013, softwareengineering.stackexchange.com/questions/188352/how-to-choose-between-lightweight-or-heavyweight-framework-for-web-application.

Lang, Mia. “Understanding Object-Relational Mapping: Pros, Cons, and Types.” AltexSoft, 11 Mar. 2021, www.altexsoft.com/blog/object-relational-mapping.

MDN Contributers. “Express Tutorial Part 4: Routes and Controllers.” MDN Web Docs, 13 Sept. 2022, https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes.

Murtaza, Alexandra. *The Benefits of Using Frameworks for Web Development*. 26 Oct. 2021, www.creative-tim.com/blog/educational-tech/benefits-frameworks-development.

Naskar, Vivek, et al. “What Are the Reasons We Use Libraries and Frameworks in Web Development - the Developer Story.” The Developer Story - Every Developer Has a Story!, 15 Sept. 2021, thedeveloperstory.com/2021/09/15/what-are-the-reason-we-use-libraries-frameworks-in-web development.

Ranjan, Ritesh. “What Is a Framework in Programming and Why You Should Use One.” *Insights - Web and Mobile Development Services and Solutions*, 7 Sept. 2022, www.netsolutions.com/insights/what-is-a-framework-in-programming.

Stalin, Lenin. “Why Spring Is a Lightweight Framework – Lenin Stalin.”. *Medium*, 9 Dec. 2021, medium.com/@leninstalinesec/why-spring-is-a-lightweight-framework-f40c36d7ad44.

“The PHP Framework for Web Artisans.” Laravel, https://laravel.com/docs/9.x/controllers.

“What Is an ORM, How Does It Work, and How Should I Use One?” Stack Overflow, 14 Aug. 2009, stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one.

“What Is Mean by a Lightweight Framework?” *Stack Overflow*, 8 Oct. 2013, stackoverflow.com/questions/19244830/what-is-mean-by-a-lightweight-framework.