

Revisiting Web Application Development with Integrated Records Management Important Aspect using Re-CRUD

¹Asyraf Wahi Anuar, ²Nazri Kama, ²Azri Azmi, ²Hazlifah Mohd Rusli

¹Faculty of Information Management,
UiTM Cawangan Johor, Kampus Segamat, Johor
Malaysia

²Advanced Informatics Department, Razak Faculty of Technology and Informatics,
Universiti Teknologi Malaysia, 54100 Kuala Lumpur,
Malaysia

Email: muham6853@uitm.edu.my, mdnazri@utm.my, azriazmi@utm.my,
hazlifah@utm.my

Received Date: 6 February 2022

Accepted Date: 26 February

Published Date: 2 April 2022

Abstract. A modern web application is plagued by a jungle of heterogeneous aspects and web technologies. Web frameworks or web application frameworks are examples of these technologies. These technologies have made it conceivable to confront verifiable issues of this WA development, for example, code automation and rapid prototyping. As the main factor for their features, both methods suggest the automation for redundant development tasks. A great analogy of a competitive and repetitive task that can be automated is the implementation of Create Read Update Delete (CRUD) operations. However, although the web application frameworks have provided a mechanism for CRUD operations, there are still many challenges that exist in integrating the other important aspects for modern data-driven WA such as searching, archiving, sharing, reporting and audit trail. So automation has not been properly faced yet. Further enhancement and manual code modification are required to improvise the half-baked generated CRUD especially in the functional integration which is time-consuming and comprise repetitive coding for each of the CRUD output. This paper presents Re-CRUD that integrates electronic records aspects into CRUD operation. Our approach is illustrated by combining a complete and robust methodology for dynamic code automation, electronic record management important aspects and Model View Controller (MVC) metaphor. A proof of concept study is carried out to demonstrate and validate the electronic record important aspect integration into CRUD operation for web application development. With the integrated important aspects in Re-CRUD, the development

process is faster and able to automatically generate more features to support modern web applications.

Keywords: web; web application framework; CRUD; web application development.

1 Introduction

Web application (WA) is increasingly prevalent in various application scope, including education, banking, communication, health, agriculture, and manufacturing. WA is being designed, developed, and deployed at a very fast pace to ensure it can cope with the rapid changing of market needs and opportunities. WA is typically developed in a multi-tiered layer architecture using several different modules and components that are very loosely related. Figure 1 shows a high-level generic architecture for most WA (Laaziri, Benmoussa, Khouli, Mohamed Larbi, & Yamami, 2019; Raunak, Kuhn, & Kacker, 2017). Efficiently developing, testing, and deploying WA remains a challenge for developers (Herrero Agustin & Del Barco, 2013; Paolone, Marinelli, Paesani, & Di Felice, 2020; Raunak et al., 2017). The rapid rate of change to web technologies highlights the need for methods that generate reusable components for WA development. Technically, WA is considered a complex software as it is composed of heterogeneous components (Herrero Agustin & Del Barco, 2013) combined to ensure the WA meets its requirements and functionality.

Considering the complexity of WA, the web application framework was invented to support WA development. The web application framework's objectives are not simply to provide utilities for WA development but also to make the development process as easy as possible (Massimo Di Pierro, 2011). Web application framework acts as a platform for Rapid Application Development (RAD) where it promotes CRUD operations, code reuse, and universality. The introduction of the web application framework has modernized the development of WA by promoting better coding management using MVC architecture, CRUD operations, rapid development concept, libraries, template, session management, and code reuse (Miles, 2016; Pitt, 2012; Prokofyeva & Boltunova, 2017). CRUD operations in WA development enable the less experienced developer to generate the skeleton of the WA (Miles, 2016).

Using CRUD operations, the developer can automate the repetitive and recurrent tasks for fundamental functions. Technically, CRUD operations in a web application framework are limited only to generating fundamental components and functions. These components and functions lack essential features that should be available in a WA, such as searching and filtering, simple authentication and authorization, integrated front-end framework, asset management, and archival features (Bartosz Porebski, Karol Przystalski, 2011; Dāsa, 2016; Golding, 2008; Kai, Omokore, & Miller, 2009; Ries & Palkó, 2019; Yu & Jin, 2018). These essential features need to be integrated after the CRUD operation is completed. Integrating the other WA features is time-consuming. It requires more technical skills and testing, which slows down the entire development process.

The CRUD operation enables the developer to generate fundamental files and codes to form a WA skeleton. Based on the database table configuration, it parses the

database schema and generates code for each table based on the MVC pattern (Giatsoglou, Koutsonikola, Stamos, Vakali, & Zigkolis, 2010). Most developers use a web application framework with CRUD operation to speed up the development process. The CRUD generator can be executed using a console framework that is accessible from the terminal Command Line Interface (CLI). Although it boosts the development process, traditional CRUD generator only generates fundamental functions, which still presents problems, namely their inadequacy to deal with form features, authentication, search, file management, and other functions and features. (Das & Saikia, 2016; Dāsa, 2016; Onesinus, 2019; Rodriguez-Echeverria, Preciado, Sierra, Conejero, & Sanchez-Figueroa, 2018). Further enhancement and manual code modification are required to improvise the half-baked generated CRUD operation, especially in the functional integration, which is time-consuming and comprises repetitive coding for each of the CRUD outputs (Rodriguez-Echeverria et al., 2018). A standalone CRUD could not satisfy the development of a decent and complex WA since it lacks integrated features to support functions such as authentication, authorization, files management, search, internationalization, form features, and statistic. (Dāsa, 2016; Jazayeri, 2007; Onesinus, 2019; Rodriguez-Echeverria et al., 2018).

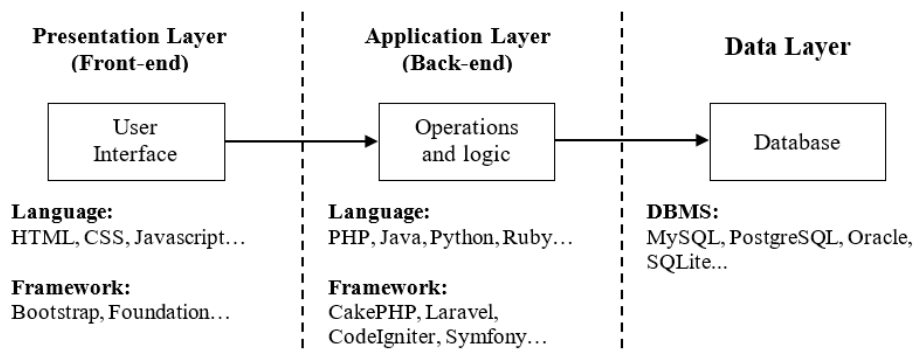


Fig. 1. Typical web application multi-layer

Based on the review of previous studies, various solutions have been proposed, for example, using plugins and third-party tools. However, most of the studies solved the problem separately, not as an integrated solution that covers all aspects of features required in a WA development. Despite offering a full-stack solution, many of the solutions are considered complex. They have a higher learning curve to be adopted as a solution, especially those requiring the use of other dependency components. As a result, the developer needs to perform significant modifications to integrate the necessary features to enhance the generated CRUD. The integration requires more time as the developer need to identify the necessary features, read and understand the documentation and specifications. The developer also needs to test the compatibility of the features before executing the integration process, which will affect the development productivity.

Problems in integrating the plugins may lead to vulnerability in the application due to the complexity of coding integration. The concept of using a framework is to build software rapidly, but if the integration of the plugin requires more time, it partially beats the purpose of using the web application framework. There is also an argument that the CRUD operation is not yet a complete solution to develop the WA as there are many redundant tasks such as repetitive codes modification for features integration after the CRUD is generated (Rodriguez-Echeverria et al., 2018). Coding these steps repeatedly takes a long time and increases development costs (Bandirmali, 2018).

In practice, features integration in a CRUD generated WA is a time-consuming process and requires multiple programming languages understanding since the features might be written in different languages. It also requires significant code enhancement to include all the necessary WA features such as authentication, search, file management, etc. This research addresses the lack of feature integration in the CRUD operation for WA development which needs further enhancement. The implementation of the CRUD operations shows that the repetitive and recurrent tasks in WA development can be systematically automated using standard coding generation and a model-driven approach (Rodriguez-Echeverria, Preciado, Rubio-Largo, Conejero, & Prieto, 2019).

2 Motivation and related works

The rapid evolution of WA technologies has led to innovative aspects to support the requirement's development. A classification of this type of aspect is presented as follows:

- Web application framework: facilitates web programming which makes it more practical and better organized in WA development and increases productivity (Prokofyeva & Boltunova, 2016).
- CRUD operation: provision of assistance in the form of code generation and basic functionalities to support the developer to accomplish the task (Noppadon & Panita, 2013). CRUD is a great technique to start a WA project as it provides automation in design patterns (Hall, Digennaro, Ward, & Havens, 2003).
- MVC: separates an application into three layers, model; view; and controller (Freeman, 2015; Hao, Zhang, & Ma, 2019; Kelly, 2019; Singh, 2016; Yin et al., 2014), which allows the developer to design, execute, and test each portion independently and keeps the code organized.
- Plugins: or plug-in, add-in, or add-on is a set of components that provides extra abilities to enhance and customize the WA functions, and it does not modify the core function or features of the WA framework (Bartosz Porebski, Karol Przystalski, 2011).
- Born-digital records: Born-digital records have no physical footprint. They began life as digital entities such as a document from word processing, images, video, 3D model, data set, and database (Kirschenbaum, 2013; Muhanna, 2018).

- Although many web technology aspects and methods have been widely adopted by developers, the requirements and functionality demanded from the WA need to be composed of heterogeneous aspects, making it complex and time-consuming. Even though the CRUD operation enables the developer to generate the WA fundamental structure and functions, it still requires major modification to ensure that the generated WA are competent to be deployed. Re-CRUD is considered in this paper as a composition of various web aspects that can be combined to shape a particular task and feature.

Web Application Framework

The web application framework (WAF) is a bunch of source code organized into a certain architecture that can be used for the rapid development of WA (Bartosz Porebski, Karol Przystalski, 2011). It can be viewed as a set of systematically organized source codes based on a certain system development architecture that can be used to support the development of dynamic WA (Dāsa, 2016; Golding, 2008). Web application framework also promotes a standard method for WA development by using Model View Controller (MVC) architecture and Rapid Application Development (RAD) method to rapidly build WA (Dennis, Wixom, Roth, Wixom, & Roth, 2014; Kelly, 2019).

The web application framework has unlocked new perspectives in terms of WA enrichment towards the development process and time, features, code organization, continuous integration, documentation, learning curve, etc. (Laaziri, Benmoussa, Khouliji, Larbi, & Yamami, 2019; Olanrewaju, Islam, & Ali, 2015; Prokofyeva & Boltunova, 2017). Since 2000, many open-source web application frameworks have been released to the public, such as CakePHP, Laravel, and CodeIgniter. These frameworks are still actively maintained by crowdsourcing initiatives. The PHP-based web application framework is among the most popular framework used in developing WA because it is dynamic, flexible, easy to use and learn (Li, Karnan, & Chishti, 2018). Moreover, the advantages of the web application framework are the elegance of codes and the ability to minimize the risk of programming errors. Web application framework conforms to the Don't Repeat Yourself (DRY) principle where all programming logic is coded only once in one place and promotes code reuse (Bartosz Porebski, Karol Przystalski, 2011). DRY principle prevents code duplication and error, which also facilitates easier maintenance of code and WA. Generally, WAF focuses on rapid development, promotes code reusability and programming best practices which, is great for developers who do not have enough knowledge or discipline to care about the quality of code. The web application framework has been widely practised by many developers to increase programming productivity as the framework is more flexible, allows rapid built of WA, MVC-based, secure and most of them are published under open-source license, which will reduce the final cost of development (Laaziri, Benmoussa, Khouliji, & Kerkeb, 2019; Olanrewaju et al., 2015; Prokofyeva & Boltunova, 2017). The advantages of web application framework adoption in WA development can be highlighted as:

Revisiting Web Application Development With Integrated Important Aspect Using Re-CRUD

- Easy to work with dynamic content for application projects such as document management, an e-commerce store for online shopping, social networking, and others.
- An easy scaling WA can grow over time without the need for large code changes.
- Encourage code reusability and modularity.
- Better code management, easy to maintain and secure.
- Rapid development concept using CRUD generator.

Create Read Update Delete (CRUD)

Create, read, update and delete are the four basic functions that manage the WA resources (Martin, 1983; Onesinus, 2019; Rodriguez-Echeverria, M. Conejero, Preciado, & Sanchez-Figueroa, 2016; Rodriguez-Echeverria et al., 2018; Watts & Jorge, 2014). The traditional CRUD application is developed using a relational database that only comprises basic components known as to create, read, update, and delete (Vohra, 2007). Computer scientists always refer to these functions by the acronym CRUD. Create is a component that allows the user or the WA itself (or both) to add a new data item into the database. Read component is used to retrieve some or all items from the database and render them into a web page. Update component enables the user or the WA to edit an existing item and have the changes stored back in the database. The last component, delete, enable the user or the WA to remove an item record from the database (McFedries, 2018).

The CRUD paradigm is common in WA development as it enables the developer to generate basic WA routine code functions such as create, read, update, delete objects and define how objects are related to each other in a WA (CakePHP, 2020; Rails Guides Team, 2015; Sinha, 2019). Thus, it enables the developer to create a quick-start application to work as the foundation of the WA solution (Golding, 2008; Stauffer, 2019). Furthermore, CRUD is a very useful time-saver as it generates the skeleton codes for the WA and enables the developer to get faster output and quickly demonstrate the basic function (input, process, and output) to the WA project stakeholder (Mark, 2013).

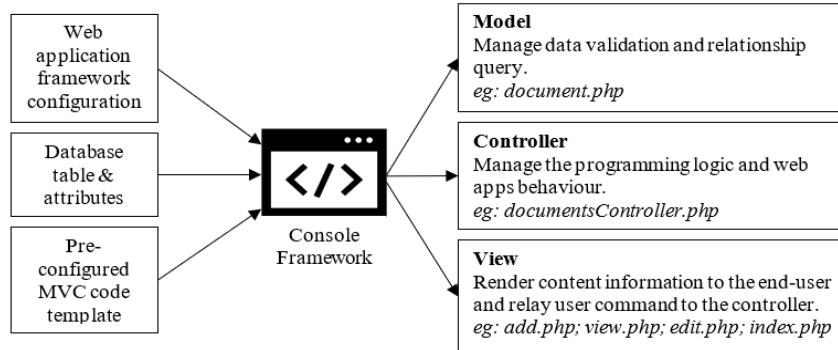


Fig. 2. CRUD operation

Figure 2 shows the CRUD operation method that leverages the pre-configured MVC template and Objects Relational Mapping (ORM). The pre-configured template is used as code mapping for the CRUD function. The pre-configured template is structured based on the MVC architecture and parted into actions and view templates (Lazetic, Savic, Vlajić, & Lazarević, 2012; McArthur, 2008). The actions template comprises CRUD actions which will form the controller, and the view template will form the user interface for the WA. The console framework will extract the database tables and use the pre-configured template to generate the CRUD function using the CLI based on the Object Relational Mapping (ORM) (Bartosz Porebski, Karol Przystalski, 2011; Mark, 2013). The output is based on the MVC architecture, which comprises the WA's model, view, and controller. Each of the CRUD acronyms can be mapped to a standard Structured Query Language (SQL) statement and Hypertext Transfer Protocol (HTTP) method to build RESTful APIs (Mike, 2017) as shown in Table I.

Table 1. CRUD operation mapping

Operation	SQL	HTTP	RESTful
Create	INSERT	PUT / POST	POST
Read (view)	SELECT	GET	GET
Update (edit)	UPDATE	PUT / POTS/ PATCH	PUT
Delete	DELETE	DELETE	DELETE

The implementation of CRUD in WA development provides a substantial productivity boost for developers (Martin, 1983; Onesinus, 2019; Rodriguez-Echeverria et al., 2016, 2018; Watts & Jorge, 2014). Using CRUD, the developer does not have to worry about many subtle details for wiring up the controller for the MVC application (Onesinus, 2019; Rodriguez-Echeverria et al., 2018). The CRUD operation is a one-time code generation process intended to jump-start the WA development (McFedries, 2018). As the WA model changes, the developer needs to manually update the controller and views. Regenerating the CRUD will drop all custom modifica-

tions, and standard coding will be applied (Onesinus, 2019; Rodriguez-Echeverria et al., 2018).

Web Application Important Features

The searching and statistic feature in a WA has become more significant due to the demand for faster data retrieval from huge data compilation. Therefore, to gain competitive advantages, the statistic component is beneficial, especially in producing reports (J. Chen, Jia, & Duan, 2011; Chu, Chao, & Yang, 2015; Daly, 2007; Dixit, 2016; Gutlić & Mujčić, 2020; Jazayeri, 2007; Wang, Kessentini, & Ouni, 2016).

The search function has also become a vital feature in any WA, as the idea of using WA is not only to systematically manage data and information but also to ensure that accurate information can be provided and retrieved when requested (Karnouskos, Da Silva, & Ilić, 2013). Working with data and information can be more difficult if no proper filtering system manages the user request (Karnouskos et al., 2013; Maskrey, 2016). Search is one of the functional components required to be present in an information system to enable fast information filtering (Felin & Kauffman, 2019; Gipp & Ebert, 2007; Kalinin, Cetintemel, & Zdonik, 2015; Porebski, Przystalski, & Nowak, 2011). However, WA created using basic generated CRUD requires manual modification to include the search function, as it is not included in the four fundamental functions of persistent operations. Search is considered as another component where the WA and database can still operate without a search function but it is a highly desirable feature to have especially in supporting data filtering (Bénel et al., 2010; Rauf and Porres, 2011; Wang et al., 2016).

The idea of CRUD is not limited to the fundamental operations only as the operations of WA today is beyond data and information storage (Daly, 2007). The WA is expected to have the analytical features as most organizations design and use WA to gain competitive advantages such as faster data processing and producing analysis reports (Yu & Jin, 2018). There are several components involved in analysis and statistic generation, which are the formulation of the data and visualization of the result that requires new functions in WA core programming (Kautz, Madsen, & Nørbjerg, 2007; Murugesan, 2007). The formulation needs to be configured in the programming logic.


```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Asyraf> cd c:/xampp/htdocs/cakephp4
PS C:\xampp\htdocs\cakephp4> bin/cake
No command provided. Choose one of the available commands.

Current Paths:

* app: src\
* root: C:\xampp\htdocs\cakephp4\
* core: C:\xampp\htdocs\cakephp4\vendor\cakephp\cakephp\

Available Commands:

App:
- console
- help

Bake:
- bake
- bake all
- bake behavior
- bake cell
- bake command
    
```

Fig. 3. CakePHP console framework using Windows PowerShell

To visualize the statistical data, it is recommended to use a data visualizer such as charts that represent the statistical table (Da Rocha, 2019; Kumar & Gupta, 2019). The integration of analytics features in the CRUD generation process will be an advantage as modifying code manually after the CRUD has been generated is time-consuming. (Daly, 2007; Kessentini, Wang, Dea, & Ouni, 2017). To include the search and analytics features, the CRUD generator engine needs to be modified, and the proper variables from the database table need to be configured. The database table will be extracted for the CRUD operations and generated based on the MVC pattern using the console framework (Massimo Di Pierro, 2011; Miles, 2016).

Console framework is considered as an engine for CRUD operation in WAF which will execute the task of generating the files and coding for the CRUD function based on the WAF architecture (CakePHP, 2020; David, 2016; McArthur, 2008; Pitt, 2012; Zdenek, 2014). There are two types of CRUD generator, i) Command Line Interface (CLI) based; ii) web-based generator. The console uses a dispatcher-type setup to load the shell script and parameter (Golding, 2008; McArthur, 2008). With the use of the console framework, the developer can generate the basic skeleton of the WA based on the MVC architecture (CakePHP, 2020). The console framework is not limited to CRUD generation commands only. Other commands are also available such as migration, routes, internationalization, cache, and others (Golding, 2008).

Figure 3 shows the CakePHP console framework CLI executed in Windows PowerShell. Technically, the console framework will retrieve all tables from the connected database, and based on the setup conventions, the console framework will suggest the table association and relevant validation for the input field (Golding, 2008; Porebski et al., 2011). Although it only generates fundamental functions (add, edit, read, delete), it is a rapid method for the developer to generate the skeleton of the WA and start to focus on other important features and functionalities

Model View Controller (MVC)

The Web application framework's primary concern is to assist the developer in rapidly building a WA and better organization of resources (Prokofyeva & Boltunova, 2017). Most web application frameworks are based on MVC architecture to isolate the data model and business rule from the user interface (UI) (Kelly, 2019; Pop & Altar, 2014). Using MVC architecture is generally considered a good practice as it modularizes code, promotes code reuse, systematic code and logic management, and allows multiple interfaces to be applied (ÉricQuinton, 2017; Singh, 2016). The Model-View-Controller (MVC) architecture started in the 1980s. It was initially used in desktop application design and later adopted in WA. MVC architecture separates an application into three layers, model; view; and controller (Freeman, 2015; Hao et al., 2019; Kelly, 2019; Singh, 2016; Yin et al., 2014), which allows the developer to design, execute and test each portion independently and while keeping the code organized.

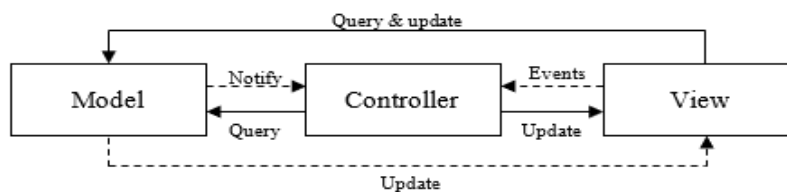


Fig. 4. MVC architecture (Pitt, 2012)

The model layer provides data maintenance and associate logic to the view, which represents knowledge, information, and a set of rules that handle updates and access to the information (Daoudi, ElBoussaidi, Moha, & Kpodjedo, 2019). Most of the MVC frameworks use the model layer to manage data validation and relationship query. View layer will render content information to the end-user and relay the user command to the controller. The rendered content information will be presented in a systematic UI to make the content more presentable and usable (Dey, 2011). The controller layer is the request handler which will render command from the end-user to the information system (IS) by providing an appropriate response to the request (Frank, Kevin, & Douglas, 2007). The controller will manage the programming logic and WA behaviour. Figure 4 shows the MVC architecture process in handling the WA request.

Isolation of functional modules makes the development process and maintenance easier for the developer to understand and modify each module without knowing everything about the application (Krasner & Pope, 1988; Pop & Altar, 2014). Therefore it also promotes more efficient collaboration among the developer (McArthur, 2008). The dominance of MVC architecture in WA and mobile application development continues to gain popularity among developers (Daoudi et al., 2019). MVC architecture speeds up the WA development process (Ivanovich, Vladimirovich, Victorovich, Viktorovich, & Vitalievna, 2019) with the flexibility to integrate with other programming languages and modules (Dey, 2011)

Web Application Framework Plugins

Web plugin (or plug-in, add-in, or add-on) is a set of components that provides extra abilities to enhance and customize the WA functions. It does not modify the core function or features of the WA framework (Bartosz Porebski, Karol Przystalcki, 2011). WA plugin contains programming logic that serves to extend the capabilities of the web application framework and it involves modification of framework core functionalities. The plugin depends on the service provided by the host and does not usually work by itself (Ratnayake, 2015). Conversely, the WA operates independently of the plugins, making it possible for the developer to enhance and update plugins dynamically without changes to the WA core.

A plugin usually is designed to fulfil single function features for enhancement. Most web plugins can be integrated and run independently with various web application frameworks such as CakePHP, Laravel, CodeIgniter. They can be released as an open-source project so that developers can leverage the plugin features into their project and improvise it (Golding, 2008; Onesinus, 2019; Sinha, 2019). Adopting a plugin in WA development may make it easier for the developer to enhance the WA features. However, it can be complicated if there is a conflict between plugins when two or more plugins interfere with one another, causing changes to their expected behaviour. It is highly challenging to identify plugin conflicts due to the exponential explosion of the plugins mixture (H. V. Nguyen, Kästner, & Nguyen, 2014)

Managing Born-Digital Records

Most of the WA is designed to have the ability to capture data, process it into valuable information and stored it as a record to make it available when it is required. Most of the WA users are not aware that the born-digital records is easy to produce but it is hard to maintain due to the nature of the digital environment and technologies where the data, information and records are easier to be generated and stored but it may lead to the information explosion due to no proper mechanism in managing the born-digital records. Since the early 1980s, system applications such as Electronic Document Management Systems (EDMS), Electronic Records Management Systems (ERMS), Electronic Document and Records Management Systems (EDRMS) have been used to manage digital content (Katu, 2012, 2016; L. T. Nguyen, Swatman, & Fraunholz, 2007). The management of electronic records involves several procedures which can be summarized as the preservation of digital records, retention schedules, and appraisal process where the action will be taken to the records, either it will be archived or disposed (Eastwood, 2004; Eastwood, Katu, & Killawee, 2000; Maroye et al., 2017; Niu, 2014). The features of born-digital records management can be integrated into a WA at the development stage. The lifecycle of digital records begins at the first stage of its creation. Therefore, the strategy to follow must be defined from these early stages onward while taking into account the process and procedure that needs to be followed (Maroye et al., 2017).

3 Electronic Record Important Aspects in Web Application

The generated CRUD is only limited to a WA's fundamental function, which includes the create, edit, read and delete functions. Besides the basic functions, WA requires more complex features such as authentication, search, retention, appraisal, archived, disposed of, sharing, audit trail, and reporting (Bartosz Porebski, Karol Przystalski, 2011; Dāsa, 2016; Golding, 2008; Kai et al., 2009; Ries & Palkó, 2019; Yu & Jin, 2018). Although some of the features are published as a plugin, it requires the developer to identify suitable plugins and have the necessary technical skills to integrate the features into the CRUD output. The weakness of this process is that it is time-consuming, prone to errors in code modification, WA functionality, and bad performance. The aspects can be explained as follows:

- Inventory - A descriptive listing of each record series or system, together with an indication of the location, access, and other pertinent data (Ira A. Penn & Gail B. Pennix, 2017; Judith Read & Mary Lea Ginn, 2016; Patricia C. Franks, 2013).
- Retention schedule - Lists how long each record series must be kept (the retention period), when the retention period starts (the cut-off), and the proper way to dispose of the record once retention is met (the disposition method) (Diamond, 2017; Judith Read & Mary Lea Ginn, 2016; UK National Archive, 2012).
- Appraisal - The process of determining the archival value and ultimate disposition of records. Appraisal decisions are informed by several factors, including the historical, legal, operational, and financial value of the records (Harvey & Thompson, 2010; Lee, 2019; Vellino & Alberts, 2016).
- Disposition - The process of destroying records or the transfer of records to another entity (most commonly as an Archive) for permanent preservation (Crockett, 2011; IRMT, 2009).
- RBAC - Provides a role-based access control mechanism in the electronic record management application to offer protection from unauthorized access. Authenticated users with different roles have different authorization or access to the records (Guo, Li, Nejad, & Shen, 2019; Lo, Wu, & Chuang, 2017; Masenya, 2020).
- Search - Enables the user to locate and retrieve records based on specific metadata, word, or phrases. It is a vital function in any WA as it enables fast data retrieval via the search parameter (Haggerty & Scott, 2019; Oladejo & Hadžidedić, 2021; Pauline, Shelda, & Peter, 2013).
- Audit trail - Provides log tracking for any changes to the electronic records to ensure validity and integrity (Lemieux, 2016; Namukasa, 2017).
- Archiving - Transfer and store the valuable records into a repository which makes them non-active but still accessible through the system. It also helps to reduce the cluttered old and non-active records from the system (Broussard & Boss, 2018; Pappel, Butt, Pappel, & Draheim, 2021).

- Sharing - Provides sharing features by sending links to other parties (public access or restricted access depends on the policy) (L. Chen, Lee, Chang, Choo, & Zhang, 2019; International Council on Archives, 2013; Lengstorf & Wald, 2016; The National Archives United Kingdom, 2017).
- Reporting - Reporting provides a summary related to the current status of records such as total records, active, inactive, the total required appraisal attention, and others (L. Chen et al., 2019; Lengstorf & Wald, 2016).

4 Methodology

The present WA is constructed based on a scheme consisting of three sub-layer, i) presentation layer; ii) application layer, and; iii) data layer. The traditional CRUD operation only comprises basic components known as create, read, update and delete (Martin, 1983; Vohra, 2007). Technically, most web application frameworks such as CakePHP, Laravel, Symfony, and FuelPHP have a build-in CRUD generator accessible through console framework, which can be executed using CLI. A review of the official documentation shows that the execution of CRUD operation using a CLI-based engine is more practical and productive due to the less configuration required, which involves more time and procedures (CakePHP, 2020; Laravel, 2019; Symfony, 2019). The CLI-based generator offers comprehensive and complex features, yet it is still simple and easy to use. It only requires a single command to execute the CRUD operation, and it will generate the files on the respective folder based on the MVC architecture (Bartosz Porebski, Karol Przystalski, 2011; CakePHP, 2020; Golding, 2008; Laravel, 2019; Symfony, 2019). CakePHP console framework CLI command is straightforward and has a gentle learning curve compared to others (Dāsa, 2016; Mark, 2013).

Combining the searching and filtering, authentication and authorization, front-end framework, and archival features into the console framework will enable the CRUD generator to generate more functions in the CRUD output. The integration process requires the restructuring of the CRUD (re-CRUD) operations and procedures. Figure 5 provides a simplified overview of the proposed design based on the CRUD evolution framework using the console framework. The traditional CRUD introduced in 1983 was mainly used for database abstraction. From there, it evolved into other application development segments such as WA development using the CakePHP framework. The console framework is used to execute the files and code generation to make the CRUD operation generation process more systematic and easier. The proposed Re-CRUD will stitch together all of the aforementioned important aspects into the console framework to provide a comprehensive solution for CRUD operation for WA development.

Revisiting Web Application Development With Integrated Important Aspect Using Re-CRUD

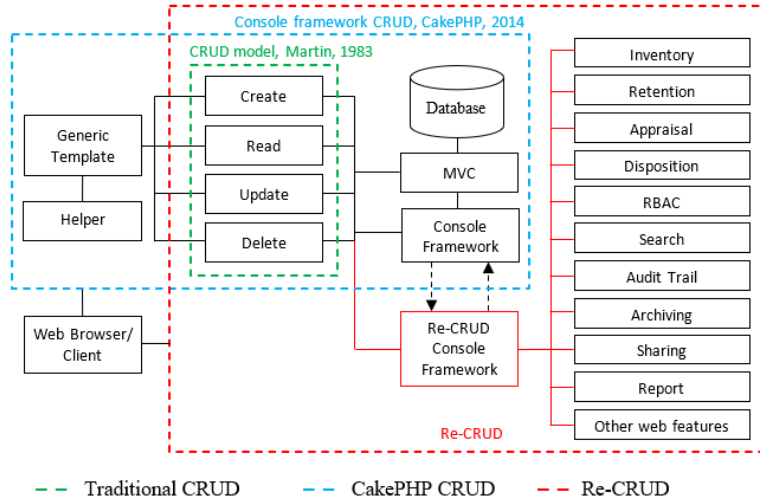


Fig. 5. Proposed Re-CRUD

The integration process is based on the Agile, iterative methodology as it provides a combination of iterative and incremental development models (Larman & Basili, 2003). In the Agile, iterative model, integration can be broken into small incremental builds. It is not developed as a complete product on the go. The development model is essential as several important aspects will be integrated into the console framework. The features integration is completed one by one and test based on priority. The Agile, iterative method is adopted to ensure the integration process undergoes a systematic method that provides more flexibility in adapting changes, adding new important aspects easily, and testing is executed in every increment. The Agile iterative model, as shown in Figure 6 will be used considering the aforementioned development concept.

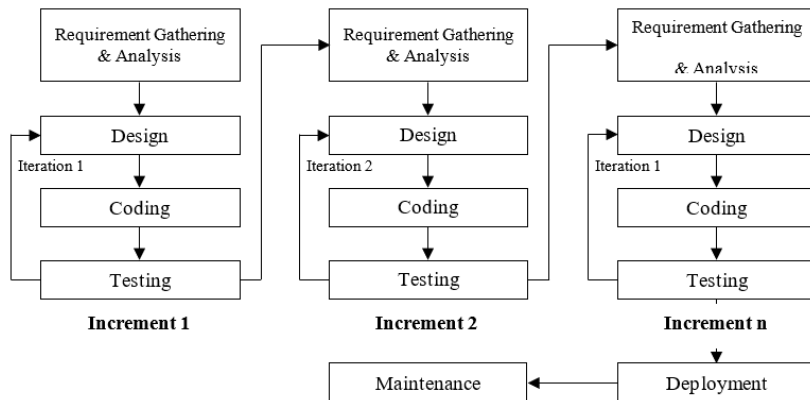


Fig. 6. Agile iterative model (Larman & Basili, 2003)

As illustrated in Figure 6, the design procedure consists of four (4) processes that are grouped into several iterations that repeat the same process and is explained below in Table 2:

Table 2. Agile iterative process

Process	Details
Requirement gathering & analysis	Each of the important aspects is downloaded and analyzed for integration purposes. All necessary details will be extracted from the host web or git documentation for analysis.
Design	Define the important aspect integration method to the console framework and prepare the integration environment.
Coding	Integrate identified important aspects for console framework enhancement. The enhancement uses PHP language, and some modifications to the core files are made due to changes in code structure.
Testing	The important aspect is integrated with the console framework and executed using CLI for re-CRUD operation. The testing covers the files and code generation together with the functionalities.
Iteration	Every identified important aspect will undergo the process, and if in the testing phase it appears to have a problem, the development needs to be re-evaluated and modified.
Increment	In every increment, a different feature will be integrated into the console framework. This ensures that every feature is appropriately integrated and the console framework produces a clean code design.
Deployment	The completed enhanced console framework is properly arranged and deployed to generate re-CRUD output.
Maintenance	Update the integrated features and console framework to ensure it is up to date.

With the integration of the aforementioned WA important aspect into Re-CRUD, more functions are provided, as shown in Table III.

Table 3. Re-CRUD operations

Operation	Method	Description
Create	add	Creates a service data objects and its descendants
Read	view	Retrieve a single dataset by the unique identifier (primary key/slug)
Update	edit	Update a dataset to the database and its related objects
Delete	delete	Delete a dataset from the database and its related objects
Search	index; search	Construct a list of data; find and return a list of a dataset that meet the specified search criteria
Retention	index_retention	List data that have passed the retention date
Appraisal	appraisal	The process of appraising the data
Disposition	disposed	The process of disposing of the data
RBAC	Authentication authorization	Authenticate user with username and password. Each account has specific roles, e.g., administrator, moderator, and user
Audit Trail	auditLog	Log all activities related to data, create, modify, dispose, etc.
Archive	archived	Set records as inactive and permanently stored for future references
Sharing	share	QR code and social media sharing links
Report	report	Statistic and data reporting

These enhancements present some noteworthy differences from the traditional CRUD generator approach, namely, i) it integrates multi-features important aspects of WA; ii) it provides more automation for important aspect integration; iii) it improve the usability of the console framework for CRUD operation and; iv) faster development process.

5 Implementation

Proof of concept study is carried out to demonstrate and validate the important aspects of integration in WA development. A document repository is generated using Re-CRUD to implement the integrated features. The document repository contains the document title, author, year of publication, and notes. Figure 7 shows the index of the document repository page, which lists all the registered documents together with the search function. Figure 8 shows the document form that is used to register new data together with the retention date. The retention date will determine the duration of the document is active before it is transferred to the retention list for appraisal purposes. This ensures that outdated or inactive documents are properly managed where the appraisal process will decide whether the inactive documents will be disposed of or archived for future references. Figure 9 is the reporting page which includes the statistic report of the document registration.

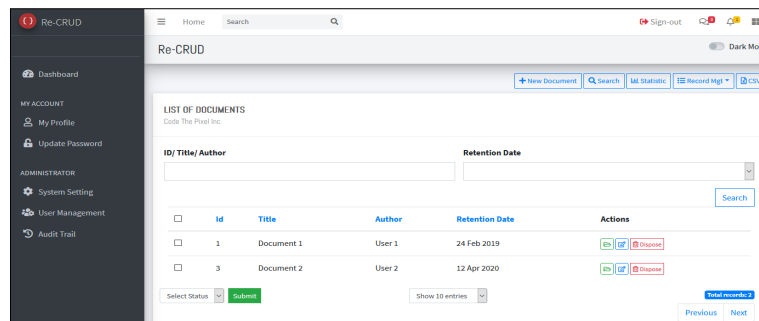


Fig. 7. Document list with search features

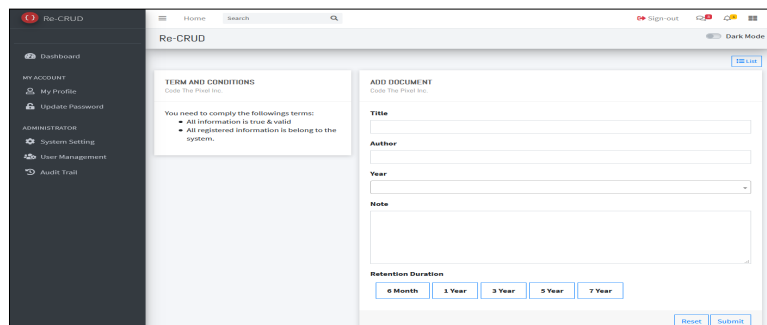


Fig. 8. Document form with retention duration

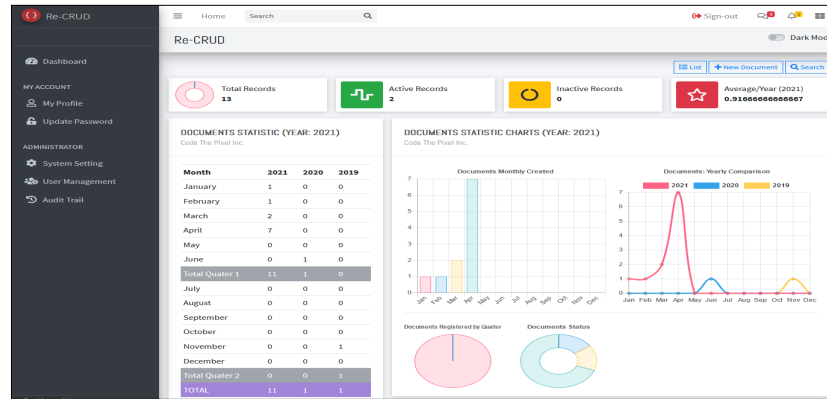


Fig. 9. Document report and statistic

6 Conclusion

The CRUD operation is aimed to speed up the development process through code automation. However, the generated components and features from the CRUD operation is limited and required the developer to integrate other important features to support the web application requirement. Generally, the process is time-consuming, comprise repetitive coding, requires more technical skills and testing, which will slow down the entire development process. Furthermore, the use of plugins to enhance WA features can be complicated if there is a conflict between plugins. Re-CRUD provides integrated electronic record aspects which include, search, retention, appraisal, disposition, RBAC, audit trail, archive, sharing and reporting features. These features is in line with most web application requirements eg: searching features is used to filter the content or data from the web application storage; archiving is to maintain the inactive records to ensure systematic data and storage management. The integrated features also support the current modern web application requirement. Furthermore, the integrated important aspects can reduce the integration error and cut the repetitive task for the developer to codes similar features to other tables.

The proof of concept has shown the electronic records important aspects integration can be successfully achieved through the CRUD operation. The generated web application comprises more features compared to the conventional CRUD output which is limited to the fundamental functions only. Using the ApacheBench, to measure response time and content delivery time in different environments. The test ware designed to determine the feasibility of Re-CRUD as a production-level solution for web application development. We found that Re-CRUD does not significantly affect the content server performance when it is performing data transactions (saved and retrieved) even though it has a significant number of integrated features. the content server performance is 0.078 seconds per resource requested when the server is under load and 0.15 to 0.25 seconds when the requested resource has many embedded components

Though the integration of important aspects of WA in Re-CRUD operation is fairly new, the idea to have an integrated search and filtering, authentication and authorization, data archival and backup together with the front-end framework will be very helpful for a developer that always needs a faster solution for their development process. In this digital age, information and systems are vital, and everything needs to be rapid and effective to ensure competency among each other. Today, in the WA development discipline, many WAF has been produced to cater to the needs of developer and business entity for them to generate WA rapidly which is commonly used to manage data and information. Technically, this research will also provide the best practice for multi-features code generation and motivate more developers to be more innovative and able to successfully execute their WA development idea. Developers who need to develop a WA may consider the Re-CRUD as the method for them to generate the WA. Future work is needed to determine if Re-CRUD is relevant to be applied to various types of web application development and functional integration can be revised based on the web application requirements. The universality of the Re-CRUD is not limited to expert WA developer-only, but it can be adopted by other profiles of developers with basic skills in programming and WA development.

Acknowledgement

The study is financially supported by the Fundamental Research Grant Scheme (Vote No. R.K130000.7856.5F415) awarded by the Ministry of Higher Education of Malaysia (MOHE) and Universiti Teknologi Malaysia.

References

- Bandirmali, N. (2018). mtCMF: A novel memory table based content management framework for automatic website generation. *Computer Standards and Interfaces*, 58, 43–52. <https://doi.org/10.1016/j.csi.2017.12.002>
- Bartosz Porebski, Karol Przystalski, L. N. (2011). *Building PHP applications with Symfony, CakePHP, and Zend framework* (1st ed.). Indianapolis: Wiley Publishing.
- Broussard, M., & Boss, K. (2018). Saving Data Journalism: New strategies for archiving interactive, born-digital news. *Digital Journalism*, 6(9), 1206–1221. <https://doi.org/10.1080/21670811.2018.1505437>
- CakePHP. (2020). CakePHP Cookbook Documentation. In *CakePHP 4* (4th ed.). Retrieved from https://book.cakephp.org/4/_downloads/en/CakePHPCookbook.pdf
- Chen, J., Jia, J., & Duan, L. (2011). DOM semantic expansion-based extraction of topical information from web pages. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6988 LNCS(PART 2), 343–350. https://doi.org/10.1007/978-3-642-23982-3_42
- Chen, L., Lee, W. K., Chang, C. C., Choo, K. K. R., & Zhang, N. (2019). Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems*, 95, 420–429. <https://doi.org/10.1016/j.future.2019.01.018>
- Chu, W. C.-C., Chao, H.-C., & Yang, S. J. H. (2015). Intelligent Systems and Applications. *Proceedings of the International Computer Symposium*, 18(2), 9–11. IOS Press.

- Crockett, M. (2011). *User Guide to Retention and Disposal Schedules Council of Europe Records Management Project*.
- Da Rocha, H. (2019). *Learn Chart.js: Create interactive visualizations for the Web with Chart.js 2*. Packt Publishing Ltd.
- Daly, L. (2007). *Next-generation web frameworks in Python*. O'Reilly Media, Inc.
- Daoudi, A., ElBoussaidi, G., Moha, N., & Kpodjedo, S. (2019). An exploratory study of MVC-based architectural patterns in Android apps. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing - SAC '19*, 1711–1720. <https://doi.org/10.1145/3297280.3297447>
- Das, R., & Saikia, L. P. (2016). Comparison of Procedural PHP with Codeigniter and Laravel Framework. *International Journal of Current Trends in Engineering & Research (IJCTER)*, 2(6), 42–48. Retrieved from <http://www.ijcter.com>
- Dāsa, R. (2016). Learn CakePHP. In *Learn CakePHP*. <https://doi.org/10.1007/978-1-4842-1212-7>
- David, S. (2016). Debugging. In *Learning PHP*. Retrieved from <https://www.oreilly.com/library/view/learning-php/9781491933565/ch12.html>
- Dennis, A., Wixom, B. H., Roth, R. M. (Roberta M., Wixom, B. H., & Roth, R. M. (2014). *Systems analysis and design* (5th ed.). United States of America: John Wiley & Sons Ltd.
- Dey, T. (2011). A Comparative Analysis on Modeling and Implementing with MVC Architecture. *International Journal of Computer Applications®*, (Mvc), 44–49. <https://doi.org/10.1.1.472.1591>
- Diamond, M. (2017). How to Implement a Record Retention Schedule for Electronic and Other Records. Retrieved February 20, 2021, from Association of Corporate Counsel (ACC) website: <https://www.acc.com/resource-library/how-implement-record-retention-schedule-electronic-and-other-records#>
- Dixit, B. (2016). Different Methods of Search and Bulk Operations. In *Elasticsearch Essentials* (pp. 143–160). Packt Publishing.
- Eastwood, T. (2004). Appraising digital records for long-term preservation. *Data Science Journal*, 3, 202–208. <https://doi.org/10.2481/dsj.3.202>
- Eastwood, T., Katuu, S., & Killawee, J. (2000). *Appraisal of Electronic Records: A Review of the Literature in English*.
- ÉricQuinton. (2017). *Using the MVC Model to Structure the Application*. <https://doi.org/10.1016/B978-1-78548-228-1.50006-0>
- Felin, T., & Kauffman, S. (2019). The Search Function and Evolutionary Novelty. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3468246>
- Frank, B., Kevin, H., & Douglas, C. S. (2007). *Pattern-Oriented Software Architecture* (4th ed.). Chichester: John Wiley & Sons Ltd.
- Freeman, A. (2015). The Model/View/Controller Pattern. In *Pro Design Patterns in Swift* (pp. 527–552). https://doi.org/10.1007/978-1-4842-0394-1_27
- Giatsoglou, M., Koutsonikola, V., Stamos, K., Vakali, A., & Zigmolis, C. (2010). Dynamic code generation for cultural content management. *Proceedings - 14th Panhellenic Conference on Informatics, PCI 2010*, 21–24. <https://doi.org/10.1109/PCI.2010.35>
- Gipp, T., & Ebert, J. (2007). Functional web applications. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4607 LNCS, 194–209. https://doi.org/10.1007/978-3-540-73597-7_16
- Golding, D. (2008). Beginning CakePHP: From novice to professional. In *Beginning CakePHP: From Novice to Professional*. <https://doi.org/10.1007/978-1-4302-0978-2>

Revisiting Web Application Development With Integrated Important Aspect Using Re-CRUD

- Guo, H., Li, W., Nejad, M., & Shen, C. C. (2019). Access control for electronic health records with hybrid blockchain-edge architecture. *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*, 44–51. <https://doi.org/10.1109/Blockchain.2019.00015>
- Gutlić, A., & Mujčić, E. (2020). Intelligent Web Application for Search of Restaurants and Their Services. In *Networks and Systems* (Vol. 83, pp. 452–469). https://doi.org/10.1007/978-3-030-24986-1_36
- Haggerty, K. C., & Scott, R. E. (2019). Do, or Do Not, Make Them Think?: A Usability Study of an Academic Library Search Box. *Journal of Web Librarianship*, 13(4), 296–310. <https://doi.org/10.1080/19322909.2019.1684223>
- Hall, R. H., Digennaro, A., Ward, J., & Havens, N. (2003). *Usability Assessment Of A Web-Based Learning System For Teaching Web Development: A Progressive Scaffolding Approach*.
- Hao, L., Zhang, J., & Ma, X. (2019). Design and Implementation of Simulation Training System Based on MVC Architecture. *IOP Conference Series: Materials Science and Engineering*, 563, 52043. <https://doi.org/10.1088/1757-899X/563/5/052043>
- Harvey, R., & Thompson, D. (2010). Automating the appraisal of digital materials. *Library Hi Tech*, 28(2), 313–322. <https://doi.org/10.1108/07378831011047703>
- Herrero Agustin, J. L., & Del Barco, P. C. (2013). A model-driven approach to develop high performance web applications. *Journal of Systems and Software*, 86(12), 3013–3023. <https://doi.org/10.1016/j.jss.2013.07.028>
- International Council on Archives. (2013). *Principles and functional requirements for records in electronic office environments*. Retrieved from https://www.ica.org/sites/default/files/9_Recordkeeping_Requirements_for_Database_Based_Business_Systems.pdf
- Ira A. Penn, & Gail B. Pennix. (2017). Records inventory. In *Records Management Handbook* (2nd ed.). New York: Routledge.
- IRMT. (2009). Managing the Creation, Use and Disposal of Electronic Records. *International Record Management Trust*.
- Ivanovich, B. V., Vladimirovich, B. V., Victorovich, N. F., Viktorovich, B. V., & Vitalievna, A. L. (2019). Using MVC pattern in the software development to simulate production of high cylindrical steel ingots. *Journal of Crystal Growth*, 526, 125240. <https://doi.org/10.1016/j.jcrysgro.2019.125240>
- Jazayeri, M. (2007). Some trends in Web application development. *FoSE 2007: Future of Software Engineering*, 199–213. <https://doi.org/10.1109/FOSE.2007.26>
- Judith Read, & Mary Lea Ginn. (2016). Electronic Records Management. In *Records Management* (10th ed.). Boston: Cengage Learning.
- Kai, C., Omokore, J., & Miller, R. K. (2009). Practical CakePHP Projects. In *Practical CakePHP Projects*. <https://doi.org/10.1007/978-1-4302-1579-0>
- Kalinin, A., Cetintemel, U., & Zdonik, S. (2015). Searchlight: Enabling integrated search and exploration over large multidimensional data. In *Proceedings of the VLDB Endowment* (Vol. 8, pp. 1094–1105). <https://doi.org/10.14778/2794367.2794378>
- Karnouskos, S., Da Silva, P. G., & Ilić, D. (2013). Developing a web application for monitoring and management of Smart Grid neighborhoods. *IEEE International Conference on Industrial Informatics (INDIN)*, 408–413. <https://doi.org/10.1109/INDIN.2013.6622919>
- Katuu, S. (2012). Enterprise content management (ECM) implementation in South Africa. *Records Management Journal*, 22(1), 37–56. <https://doi.org/10.1108/09565691211222081>
- Katuu, S. (2016). Managing digital records in a global environment: A review of the landscape

- of international standards and good practice guidelines. *Electronic Library*, Vol. 34, pp. 869–894. <https://doi.org/10.1108/EL-04-2015-0064>
- Kautz, K., Madsen, S., & Nørbjerg, J. (2007). Persistent problems and practices in information systems development. *Information Systems Journal*, 17(3), 217–239. <https://doi.org/10.1111/j.1365-2575.2007.00222.x>
- Kelly, S. (2019). Model View Controller. In *Python, PyGame, and Raspberry Pi Game Development* (pp. 213–231). https://doi.org/10.1007/978-1-4842-4533-0_17
- Kessentini, M., Wang, H., Dea, J. T., & Ouni, A. (2017). Improving Web Services Design Quality Using Heuristic Search and Machine Learning. *Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017*, 540–547. <https://doi.org/10.1109/ICWS.2017.63>
- Kirschenbaum, M. (2013). The .txtual Condition: Digital Humanities, Born-Digital Archives, and the Future Literary. *Digital Humanities Quarterly*, 007(1).
- Krasner, G. E., & Pope, S. T. (1988). A Cookbook for Using the Model-view Controller User Interface Paradigm in Smalltalk-80. *J. Object Oriented Program.*, 1(3), 26–49. Retrieved from <http://dl.acm.org/citation.cfm?id=50757.50759>
- Kumar, S., & Gupta, S. (2019). Development of Decision Support System for Promotion Assessment. *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 1169–1173.
- Laaziri, M., Benmoussa, K., Khouliji, S., & Kerkeb, M. L. (2019). A Comparative study of PHP frameworks performance. *Procedia Manufacturing*, 32, 864–871. <https://doi.org/10.1016/j.promfg.2019.02.295>
- Laaziri, M., Benmoussa, K., Khouliji, S., Larbi, K. M., & Yamami, A. El. (2019). A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering*, 9(1), 704–712. <https://doi.org/10.11591/ijece.v9i1.pp.704-712>
- Laaziri, M., Benmoussa, K., Khouliji, S., Mohamed Larbi, K., & Yamami, A. El. (2019). Analyzing bootstrap and foundation front-end frameworks: a comparative study. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1), 713. <https://doi.org/10.11591/ijece.v9i1.pp713-722>
- Laravel. (2019). Laravel - The PHP Framework For Web Artisans. Retrieved July 14, 2020, from LARAVEL website: <https://laravel.com/docs/8.x/controllers>
- Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, Vol. 36, pp. 47–56. <https://doi.org/10.1109/MC.2003.1204375>
- Lazetic, S., Savic, D., Vlajić, S., & Lazarević, S. (2012). A Generator of MVC-based Web Applications. *World of Computer Science and Information Technology Journal*, 2, 2221–2741.
- Lee, C. A. (2019). Computer-Assisted Appraisal and Selection of Archival Materials. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2721–2724. <https://doi.org/10.1109/BigData.2018.8622267>
- Lemieux, V. L. (2016). Trusting records: is Blockchain technology the answer? *Records Management Journal*, 26(2), 110–139. <https://doi.org/10.1108/RMJ-12-2015-0042>
- Lengstorf, J., & Wald, K. (2016). *Pro PHP and jQuery*. <https://doi.org/10.1007/978-1-4842-1230-1>
- Li, X., Karnan, S., & Chishti, J. A. (2018). An empirical study of three PHP frameworks. *2017 4th International Conference on Systems and Informatics, ICSAI 2017, 2018-Janua*, 1636–1640. <https://doi.org/10.1109/ICSAI.2017.8248546>
- Lo, N. W., Wu, C. Y., & Chuang, Y. H. (2017). An authentication and authorization mechanism for long-term electronic health records management. *Procedia Computer*

Revisiting Web Application Development With Integrated Important Aspect Using Re-CRUD

Science, 111, 145–153. <https://doi.org/10.1016/j.procs.2017.06.021>

- Mark, R. H. (2013). *Instant CakePHP Starter*. Packt Publishing.
- Maroye, L., van Hooland, S., Aranguren Celorrio, F., Soyez, S., Losdyck, B., Vanreck, O., & de Terwangne, C. (2017). Managing electronic records across organizational boundaries: The experience of the Belgian federal government in automating investigation processes. *Records Management Journal*, 27(1), 69–83. <https://doi.org/10.1108/RMJ-11-2015-0037>
- Martin, J. (1983). *Managing the data-base environment*. USA: Prentice Hall.
- Masenyua, T. M. (2020). Application of modern technologies in the management of records in public libraries. *Journal of the South African Society of Archivists*, 53, 65–79. <https://doi.org/10.4314/jsasa.v53i1.5>
- Maskrey, M. K. (2016). Web Services. In *App Development Recipes for iOS and watchOS* (pp. 259–299). https://doi.org/10.1007/978-1-4842-1820-4_11
- Massimo Di Pierro. (2011). web2py for Scientific Applications. *Computing in Science and Engineering*, 13(2), 64–69. <https://doi.org/10.1109/MCSE.2010.97>
- McArthur, K. (2008). *Pro PHP Patterns, Frameworks, Testing and More* (1st ed.). <https://doi.org/10.1007/978-1-4302-0279-0>
- McFedries, P. (2018). *Web Coding & Development All-in-One For Dummies*. New Jersey: John Wiley & Sons, Inc.
- Mike, A. (2017). *RESTful Web Clients*. Retrieved from <http://shop.oreilly.com/product/0636920037958.do>
- Miles, M. (2016). Using web2py Python framework for creating data-driven web applications in the academic library. *Library Hi Tech*, 34(1), 164–171. <https://doi.org/10.1108/LHT-08-2015-0082>
- Muhanna, E. (2018, February 1). What does “Born digital” mean? *International Journal of Middle East Studies*, Vol. 50, pp. 110–112. <https://doi.org/10.1017/S0020743817000976>
- Murugesan, S. (2007). Web Application Development: Challenges And The Role Of Web Engineering. In *Web Engineering: Modelling and Implementing Web Applications* (pp. 7–32). https://doi.org/10.1007/978-1-84628-923-1_2
- Namukasa, J. (2017). Records management and procurement performance: A case of NAADS program in the central region of Uganda. *Records Management Journal*, 27(3), 256–274. <https://doi.org/10.1108/RMJ-04-2016-0011>
- Nguyen, H. V., Kästner, C., & Nguyen, T. N. (2014). Exploring variability-aware execution for testing plugin-based web applications. *Proceedings - International Conference on Software Engineering*, (1), 907–918. <https://doi.org/10.1145/2568225.2568300>
- Nguyen, L. T., Swatman, P. M., & Fraunholz, B. (2007). EDMS, ERMS, ECMS or EDRMS: Fighting through the acronyms towards a strategy for effective corporate records management. *ACIS 2007 Proceedings - 18th Australasian Conference on Information Systems*, 790–800. Retrieved from <http://aisel.aisnet.org/acis2007/122>
- Niu, J. (2014). Appraisal and Selection for Digital Curation. *International Journal of Digital Curation*, 9(2), 65–82. <https://doi.org/10.2218/ijdc.v9i2.272>
- Noppadon, P., & Panita, W. (2013). Development of a Ubiquitous Learning System with Scaffolding and Problem-Based Learning Model to Enhance Problem-Solving Skills and ICT Literacy. *International Journal of E-Education, e-Business, e-Management and e-Learning*, 3(3). <https://doi.org/10.7763/ijeeee.2013.v3.222>
- Oladejo, B., & Hadžidedić, S. (2021). Electronic records management – a state of the art review. *Records Management Journal, ahead-of-p*(ahead-of-print). <https://doi.org/10.1108/RMJ-09-2019-0059>
- Olanrewaju, R., Islam, T., & Ali, N. (2015). *An Empirical Study of the Evolution of PHP MVC Framework*. https://doi.org/10.1007/978-3-319-07674-4_40

- Onesinus, S. P. (2019). *Laravel CRUD with bootstrap jQuery and Mysql*. OSPT.
- Paolone, G., Marinelli, M., Paesani, R., & Di Felice, P. (2020). Automatic code generation of mvc web applications. *Computers*, 9(3), 1–29. <https://doi.org/10.3390/computers9030056>
- Pappel, I., Butt, S., Pappel, I., & Draheim, D. (2021). On the specific role of electronic document and record management systems in enterprise integration. In *Advances in Intelligent Systems and Computing* (Vol. 1184, pp. 37–51). https://doi.org/10.1007/978-981-15-5859-7_3
- Patricia C. Franks. (2013). Records Retention Strategies Inventory Appraisal Retention and Disposition. In *Records and Information Management* (pp. 84–114). American Library Association.
- Pauline, J., Shelda, D., & Peter, G. (2013). Search behaviour in electronic document and records management systems: an exploratory investigation and model. Retrieved February 21, 2021, from Information Research website: <http://informationr.net/ir/18-1/paper572.html#.YDJWS2gzaUk>
- Pitt, C. (2012). Pro PHP MVC. In *Pro PHP MVC*. <https://doi.org/10.1007/978-1-4302-4165-2>
- Pop, D.-P. P., & Altar, A. (2014). Designing an MVC model for rapid web application development. *Procedia Engineering*, 69, 1172–1179. <https://doi.org/10.1016/j.proeng.2014.03.106>
- Porebski, B., Przystalski, K., & Nowak, L. (2011). *Building PHP Applications with Symfony, CakePHP, and Zend Framework* (1st ed.). GBR: Wrox Press Ltd.
- Prokofyeva, N., & Boltunova, V. (2016). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science*, 104, 51–56. <https://doi.org/10.1016/j.procs.2017.01.059>
- Prokofyeva, N., & Boltunova, V. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science*, 104, 51–56. <https://doi.org/10.1016/j.procs.2017.01.059>
- Rails Guides Team. (2015). Getting Started with Rails — Ruby on Rails Guides. Retrieved November 28, 2019, from https://guides.rubyonrails.org/getting_started.html#getting-up-and-running-quickly-with-scaffolding
- Ratnayake, R. N. (2015). *WordPress Web Application Development* (3rd ed.). Birmingham: Packt.
- Raunak, M. S., Kuhn, D. R., & Kacker, R. (2017). Combinatorial Testing of Full Text Search in Web Applications. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, 100–107. <https://doi.org/10.1109/QRS-C.2017.24>
- Ries, T., & Palkó, G. (2019). Born-digital archives. *International Journal of Digital Humanities*, 1(1), 1–11. <https://doi.org/10.1007/s42803-019-00011-x>
- Rodriguez-Echeverria, R., M. Conejero, J., Preciado, J. C., & Sanchez-Figueroa, F. (2016). AutoCRUD - Automating IFML Specification of CRUD Operations. *Proceedings of the 12th International Conference on Web Information Systems and Technologies, 1*, 307–314. <https://doi.org/10.5220/0005923003070314>
- Rodriguez-Echeverria, R., Preciado, J. C., Rubio-Largo, Á., Conejero, J. M., & Prieto, Á. E. (2019). A Pattern-Based Development Approach for Interaction Flow Modeling Language. *Scientific Programming*, 2019, 1–15. <https://doi.org/10.1155/2019/7904353>
- Rodriguez-Echeverria, R., Preciado, J. C., Sierra, J., Conejero, J. M., & Sanchez-Figueroa, F. (2018). AutoCRUD: Automatic generation of CRUD specifications in interaction flow modelling language. *Science of Computer Programming*, 168, 165–168. <https://doi.org/10.1016/j.scico.2018.09.004>

Revisiting Web Application Development With Integrated Important Aspect Using Re-CRUD

- Singh, S. (2016). Comparative Study of MVC (Model View Controller) Architecture with respect to Struts Framework and PHP. *International Journal of Computer Science Engineering*, 5(03), 142–150. Retrieved from www.ijcse.net/docs/IJCSE16-05-03-090.pdf
- Sinha, S. (2019). Introduction to Laravel. In *Beginning Laravel* (pp. 1–10). https://doi.org/10.1007/978-1-4842-4991-8_1
- Stauffer, M. (2019). *Laravel: Up and Running: A Framework for Building Modern PHP Apps* (2nd ed.). O'Reilly Media.
- Symfony. (2019). Generating a CRUD Controller Based on a Doctrine Entity (SensioLabs Generator Bundle Docs). Retrieved July 14, 2020, from Symfony Official Documentation website: <https://symfony.com/doc/current/bundles/SensioGeneratorBundle/commands/generateDoctrineCrud.html>
- The National Archives United Kingdom. (2017). *Migrating information between records management systems*. 1–35.
- UK National Archive. (2012). *Records Management retention scheduling*. Retrieved from www.archive.official-documents.co.uk/document/cm43/4310/4310.htm
- Vellino, A., & Alberts, I. (2016). Assisting the appraisal of e-mail records with automatic classification. *Records Management Journal*, 26(3), 293–313. <https://doi.org/10.1108/RMJ-02-2016-0006>
- Vohra, D. (2007). Ruby on rails for PHP and Java developers. In *Ruby on Rails for PHP and Java Developers*. <https://doi.org/10.1007/978-3-540-73145-0>
- Wang, H., Kessentini, M., & Ouni, A. (2016). Bi-level identification of web service defects. *Lecture Notes in Computer Science*, 9936 LNCS, 352–368. https://doi.org/10.1007/978-3-319-46295-0_22
- Watts, J., & Jorge, G. (2014). *CakePHP 2 Application Cookbook*. USA: Packt Publishing.
- Yin, X., Zheng, W., Zhang, M., Zhang, J., Zhuang, G., & Ding, T. (2014). A modularized operator interface framework for Tokamak based on MVC design pattern. *Fusion Engineering and Design*, 89(5), 628–632. <https://doi.org/10.1016/j.fusengdes.2014.03.028>
- Yu, X., & Jin, Z. (2018). Web content information extraction based on DOM tree and statistical information. *International Conference on Communication Technology Proceedings, ICCT, 2017-October*, 1308–1311. <https://doi.org/10.1109/ICCT.2017.8359846>
- Zdenek, M. (2014). *PHPUnit Essential*. Birmingham, UK: Packt Publishing.