

PAPER • OPEN ACCESS

Model Framework for Development of Biodiversity Information Systems

To cite this article: Edy Budiman *et al* 2019 *J. Phys.: Conf. Ser.* **1230** 012012

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Model Framework for Development of Biodiversity Information Systems

Edy Budiman¹, Novianti Puspitasari¹, Masna Wati¹, Haviluddin¹ and Robbi Rahim²

¹ Universitas Mulawarman, Kampus Gunung Kelua Samarinda, Kalimantan Timur

² Sekolah Tinggi Ilmu Manajemen Sukma, Jln. Laksana Medan Sumatera Utara

Email: edybudiman.unmul@gmail.com

Abstract. The aim of the study was to design a framework for developing bioinformatics resource information systems using the Model-View-Controller (MVC) design pattern. Research contributions generate a framework as an approach model for the development of Biodiversity Information Systems which aims to improve computational capabilities and management of biodiversity data resources for the use of public information clusters. This product combines component capabilities (View Controller Model (MCV), Object Relational Mapping (ORM) and ICBN Nomenclature Taxonomy) with reusable resources. The results of the study have produced a special prototype in the form of a Framework in the Development of Bioinformatics resource information systems that can be accessed online in the site: <http://borneodiversity.org/index>.

1. Introduction

Over the past 20 years, more and more bioinformatics tools have covered a growing spectrum of applications, including aspects of data management for biological species, systems, and availability of structural biology [1]. The complexity of the analysis carried out in this domain makes publication, integration, and interconnection of this tool very challenging [1]. Some systems propose solutions to automate access to bioinformatics resources. Web-based user interface [1], such as the site: mybis.gov.my, catalogueoflife.org, bis.iirs.gov.in, cbd.int, biodiversity.europa.eu, gbif.org, etc.

Bioinformatics as a study of applied computational techniques for managing and analysing biological information. This field includes the application of mathematical, statistical and informatics methods to solve biological problems, especially those related to naming species, data hierarchies, and existing nomenclature. whereas, the definition of bioinformatics according to [2] as the study of informatics processes in biotic systems at various levels is only a historical game, to be replaced by the general meaning of the term as showing the development and use of computational methods for comparative analysis genome data [2].

In Indonesia, initial efforts to mainstream sustainable biodiversity management approach at global and local levels. At the global level, the United Nations Convention on Biodiversity [3]. Ratification in the form of a national strategy and action plan published in the Biodiversity Action Plan for Indonesia [4]. One of the action plans undertaken is the availability of data and information and the development of a biodiversity information system (BIS).

Programmatic programming and the object-oriented development paradigm were created to solve some problems that procedural programming cannot be overcome. This pattern stimulates



the emergence of various problem-solving patterns. Solutions for certain problems are often used repeatedly from programmers, to solve the same problem at different times. The problem-solving pattern that is then in OOP is called the Design Pattern. Almost the same as Pattern Design, Framework, made according to the needs of programmers for a set of program libraries and other components to solve programming problems faced repeatedly. Usually, the framework is based on certain Design Patterns. One design pattern that is considered most suitable for web-based software architecture is the MVC design pattern that separates software into three parts, i.e. Model, View and Controller (MVC).

The purpose of this study is to design the framework for the development of bioinformatics resource portals with the Model-View-Controller (MVC) design pattern. Research contributions generate a framework as an approach model for the development of the Bioinformatics Resource Information System which aims to improve computational capabilities and management of biodiversity data resources for the use of public information clusters.

2. Background and Development Methodology

2.1 Bioinformatics Resource Portal

Biodiversity information (bioinformatics) is very important for scientific studies/research, education, and decision-making. In the management of natural resources, information on biodiversity is needed to maintain the sustainability of species use, explore biological potential and monitor species along with ecology, policy making, and for the development of biotechnology innovations. Given the importance of physical information, it is necessary to develop computer-based technology that can manage data, information, and biodiversity knowledge properly and efficiently so that it can be used for exploration, analysis, synthesis, and interpretation of potential biodiversity riches. Previous studies on BIS development have been carried out by the authors, such as the Eloquent object-relational mapping model for biodiversity information systems [5], Biodiversity information systems: Tropical rainforest Borneo and traditional knowledge ethnic of Dayak [6], Borneo biodiversity: Exploring species endemic trees and wood characteristics [7]. Ethnobotany Database: Exploring diversity medicinal plants of Dayak tribe Borneo [8]. Database: Taxonomy of plants Nomenclature for Borneo biodiversity information system [9], [10]. Furthermore, related to network availability in meeting system requirements in the study area has been studied in [11-12].

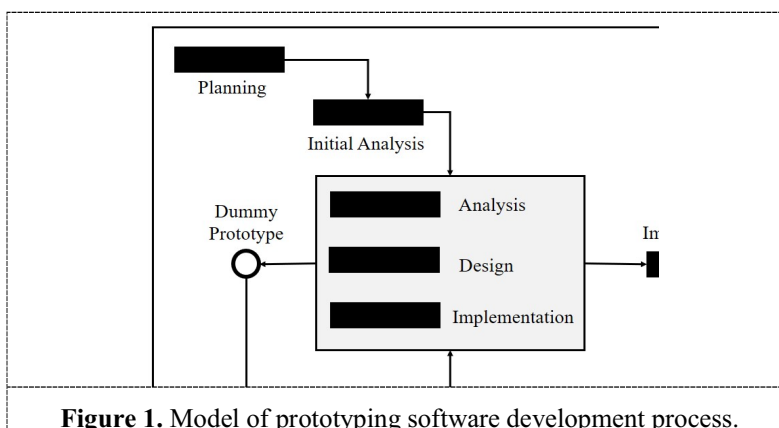
2.2 Framework Model for Web Development

One design pattern that is considered most in accordance with web-based enterprise software architecture such as the Biodiversity information system is the MVC design pattern that separates software into three parts, i.e. Model, View, and Controller. The View Controller pattern model refers to [13] defining a software architecture that separates data (Model) from view (View) and how to process it (Controller). With the separation of different aspects, it can simplify the development of complex systems because it allows to focus more on certain aspects, such as you can focus more on aspects of the display without having to rely on the logic aspects that govern the flow of business, or vice versa [13]. Previous research on PHP framework performance comparisons such as [13] have different conclusions on each framework regarding performance for a particular task, but in this study will propose a new design framework for system complexity such as BIS.

2.3 Framework Development Methods

The first basic assumption built into the framework's design is that using a web application framework in developing bioinformatics portal software can accelerate the application development process because the web application framework arranges the components that are used repeatedly better by using the right design patterns. Then this study examines how the right framework must be used for web-based applications and their applications in software development.

The framework development method used to construct this framework pattern uses a prototyping process model approaches. According to Arnowitz [14], in general, the prototyping stages are divided into four main phases [14], presented in Figure.1.



- **Planning:** this stage is needed to determine the aspects of the framework that will be made prototypes and which do not need to be made prototypes to optimize results. This phase includes needs analysis, which determines what content (library) what must be owned, and also creates workflows from the framework of the user display (programmer). The results of this first phase will be translated into specifications in the next phase.
- **Specifications:** this stage is done to determine the characteristics of the prototype to be made. The important thing at this stage is to determine who is the end user of the software? and who the stakeholders will help develop this software. In the manufacture of programming frameworks, the end user is a programmer who in fact is mostly aware of the technical process of software development.
- **Prototype Building:** In the design development process, it starts with analysing the design requirements of the framework that will be built. This analysis relates to how large and complex the system will be processed and what components are needed, the next stage is the design of the framework, this design will be used as a framework for developing complex systems such as Biodiversity on a regular basis and easily implemented. Pattern design, so that parts of the framework can be interconnected with each other to work on various processes separately based on their respective tasks, implementation of the design will be carried out for the framework that has been built.
- **Results:** Prototyping method, the result phase means the implementation phase of the prototype after the previous process through the process of review, validation, and re-execution. This phase is the longest phase compared to the previous three phases, and in this phase, stakeholders who help develop this software will play a role in testing the prototypes that have been made. The implementation and review framework is done by using it to build web-based applications. After the review is considered sufficient, in the sense that the user's needs have been fulfilled, the software can be deemed complete.

3. Results and Discussion

3.1 Results: Planning and Specifications of Bioinformatics Portal framework model

Bioinformatics resource portal framework is a collection of programming libraries in the form of resources that can be used repeatedly that are formed using certain design patterns. In the needs analysis stage, it is known what design patterns are appropriate to be applied to this framework, as well as what libraries will be used. The results of the need analysis are as in Table 1.

Table 1. Design Pattern and Library Resources used in Bioinformatics Portal Framework.

Recommendation	Type	Resized in HTML or CSS from Image Scale
MVC design pattern	Separate programs into three parts of business logic, presentation, and navigation	Separation of programs makes it easy to develop software with frequently changing needs
Blueprint CSS	CSS library to ease page layout with Grid	The use of grid layout makes it easy to place shown elements as in the print media layout
jQuery	Java-script library to simplify the functions of DOM HTML and Ajax	Is a Java-script library that emphasizes the ease of writing code for complexity functions
jQuery.ui	Additional jQuery library to make it easy to add widgets and effects	Integrated with jQuery, and also has a quite complete usage documentation
Tiny MCE	JavaScript library to modify the text-area to the rich editor for ease input	Pure java-script so it's easy to integrate with any server programming
Database and Active Record	Database abstraction to facilitate communication with various types of DBMS easily	Its use is easy and also balanced with better performance than ORM
Session and cookie library	ease the use of session and cookies that are equipped with the facility to store in the database	Special library creation for specific purposes
HTML and form library	Simplify the writing of HTML code, especially for writing forms with server-side input validation	Special library creation for specific purposes
Uploader and library file	Library to ease uploading and managing file users	Special library creation for specific purposes
URL rewrite library	Customize URLs for URL use that are more human readable	Adaptation from LARAVEL code URL that is easy to use and humanly readable

3.2 Results: Model View Controller of the Bioinformatics Resource Portal

The bioinformatics resource portal framework uses the Model View Controller (MVC) design pattern [15], which is a design pattern that divides the software structure into three major parts, i.e. programming logic, presentation display, and navigation management that connects the two parts [18]. Whereas in practice the workflow of a web application framework, not only those three parts but also involves loading libraries, translating URL addresses, and so on. End users only access one file, then this file contains a library that translates the URL address into a function that can be understood by the framework. The next step is routing from the URL address by checking it first, whether there is a cache file that matches the request, if there is then the cache file is loaded directly for displayed to the user. If there are no suitable documents, routing continues the controller request by searching for classes and methods requested by the user.

The controller is able to load all libraries in the form of libraries, additional scripts, and helper. In addition, the controller also has the task of loading the model if it needed to communicate with the database server. After working on all client requests, the controller loads the views that have created in the view to display on the user interface, but before the displayed data is stored in the caching file for faster retrieval. The last stage is the user gets the page view requested via main index.php.

3.3 Framework for Nomenclature

The framework of the plant data naming system in the database system refers to the Taxonomy of the International Code of Botanical Nomenclature [16] as in Figure. 2, is a system that shows the

level of a taxonomic unit or group and the scientific name applied to individuals of taxonomic group data on portal bioinformatics [16].

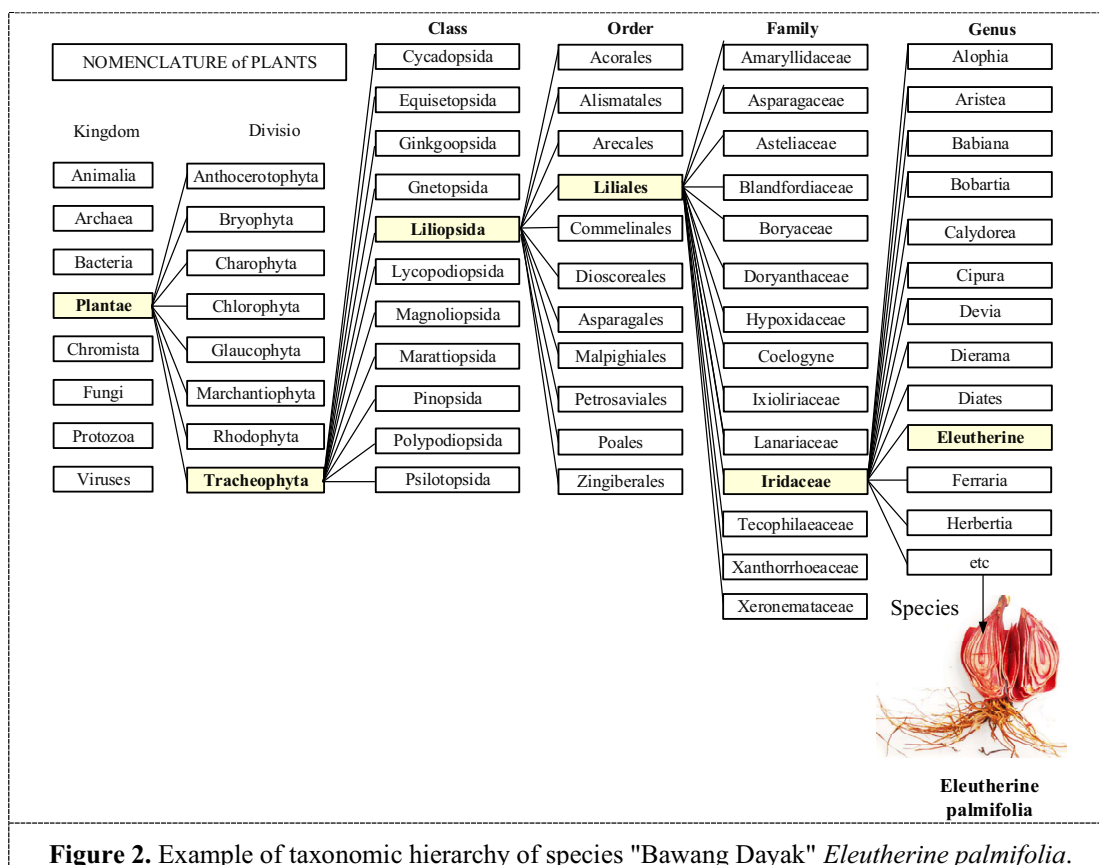


Figure 2. Example of taxonomic hierarchy of species "Bawang Dayak" *Eleutherine palmifolia*.

3.4 Results: Directory Structure for Bioinformatics Portal Framework

The directory structure is important in a framework, this is related to the design pattern that will be used and the routing of the URL address that will be used. In addition, the directory structure determines how easily the programmer finds the required file, and whether the file is at a consistent address. In this bioinformatics portal framework, in general, the folder structure is broken down into two parts, namely the system engine folder and application folder. The system engine folder is at the core of the bioinformatics portal framework, all libraries and commands that govern the system are in this folder. Only one system engine directory is needed for many application projects, because this directory only stores core libraries, while configuration and work files are all in the application directory. Whereas the application folder is the folder where we work writing the programming code that has been set in the system engine, this application directory can be duplicated as much as possible according to the number of projects done using this framework.

Figure 3, in the application directory, there is one more directory called "workspace", this directory contains directories and configuration files and the work using for to certain projects. To develop software using BBIS framework, programmers only need to deal with this directory, except for certain cases such as to replace the library database abstraction with the desired ORM. In detail, the function of each directory in the "workspace" as in Figure 3 and Table 2.

Table 2. Design Pattern and Library Resources used in Bioinformatics Portal Framework.

Model Framework	Directory Name	Directory function
	Console	Is a directory that contains classes that function to execute certain commands automatically when fulfilling a certain condition
	Exception	Is a folder that contains classes that function to handle certain errors in the program code line when the program code is being executed
	Controllers	The controller is the link between the user interface and business logic. In the controller run input validation and to change the model data into the presentation view. This controller file is doing by the programmer by extending from the existing controller class and stored in this directory.
	Middleware	A folder that contains classes that can be used as a delimiter between requests and responses given in the application business flow.
	Request	A folder containing classes that can handle user requests with various methods such as get, post, delete etc.
	Model	The model is the logic of the software, retrieves data from the database provider, or also changes the database, and sends the results to the controller. All programmers written by the programmer extend from the model class provided.
	Migration	Is a folder that contains classes for designing databases such as creating tables, arranging table relationships, and determining column and character length data types.
	Seeder	class that can be used as a tool to fill data in columns and rows of tables in a database with a certain value without having to fill in the data in the fill on the system UI.
	Views	The view file contains HTML code and other languages to form the presentation view, which will be a presented to the end users. All files are in this directory and the controller will load.
	Routes	Is a folder that contains files that function as URLs in receiving requests from users so that user requests can be processed in the application system.
	Storage	Is a folder that can be used as a place to store files uploaded by the user or file storage needed by the system
	Config	Database abstraction library provides caching result facility, which is to save query results into a file, so that later when needed to load the same query, it will only load the existing cache to improve performance. The function of this cache directory is to save the cache file.
	Cache	Database abstraction library provides caching result facility, which is to save query results into a file, so that later when needed to load the same query, it will only load the existing cache to improve performance. The function of this cache directory is to save the cache file.
	Customize URLs	Adaptation from LARAVEL code URL that is easy to use and humanly readable
	Helpers	As in the helper directory in the system engine, this directory also used to store helper. This directory allows the programmer to enter a special helper in a software.

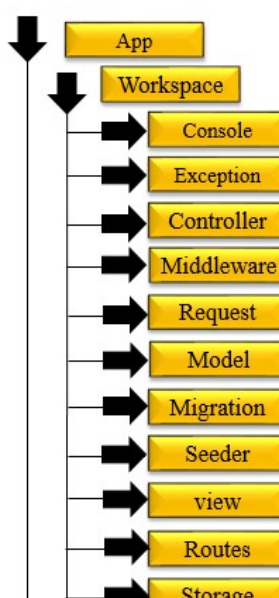


Figure 3. Directory structure

3.5 Results: Framework Implementation for Bioinformatics Portal

The system implementation phase is the implementation stage of the prototype framework and the Model View Controller which are the basic prototypes according to the functional requirements of

the system. The implementation process is carried out as the final result of system design. In the initial process, the user will make a request on the route through the browser by entering the URL.

[<https://borneodiversity.org/index/home/Bawang%20Dayak%20>]

The framework for the development of the Bioinformatics Resource Portal produced is an initial prototype (dummy prototype) with a Model-Controller View approach pattern. In general, the separation of data (model) from the view (view) and how to process it (controller). The results of the prototype portal implementation of Bioinformatics Resources are seen in Figure 4.

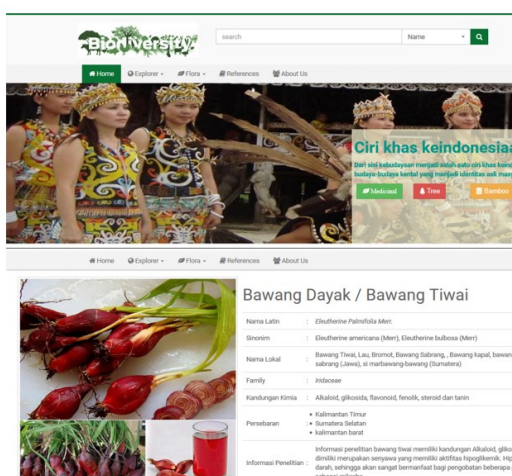


Figure 4. The View of prototype results in the Bioinformatics Resource portal.

4. Conclusion

The development of the discussed Bioinformatics Resource Portal Framework combines the ability of re-usable components, MVC, ORM and plant naming patterns in accordance with ICBN. This device can be used to help the development process of large-scale web-based software such as bioinformatics resource portals. The framework for the bioinformatics source portal developed has two directories in it, i.e. the system engine directory and application directory. The system engine directory contains a "builder" directory that contains functions for automatically creating application directories using the command line interface and "system" directory, which consists of six directories containing libraries and machines from this framework. The application directory contains one file, "workspace", this directory contains the directory and configuration files. And also the work used for a particular project. To develop software using the Portal Bioinformatics resource framework, programmers only need to deal with this directory, except for certain cases such as to replace library database abstractions with the desired ORM.

References

- [1] B. Néron, H. Ménager, C. Maufrais, N. Joly, J. Maupetit, S. Letort, S. Carrere, P. Tuffery, and C. Letondal, "Mobylye: a new full web bioinformatics framework", *Bioinformatics*, vol. 25(22), 2009, pp. 3005-3011.
- [2] P. Hogeweg, "The roots of bioinformatics in theoretical biology" *PLoS computational biology*, vol. 7(3), 2011, p.e1002021.
- [3] CBD, and UNDP, "UN Biodiversity Conference results in significant commitments for action on Biodiversity", Press Releases, the Convention on Biological Diversity and United Nations Development Programme, Dec. 2017.

- [4] Bappenas, "Indonesia Biodiversity Strategy and Action Plan 2015-2020", IBSAP Dokumen Nasional, BAPPENAS, Jakarta, 2016.
- [5] E. Budiman, M. Jamil, U. Hairah, H. Jati and Rosmasari, "Eloquent object relational mapping models for biodiversity information system," 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta Bali, 2017, pp. 1-5. doi: 10.1109/CAIPT.2017.8320662.
- [6] Dengen, N., Budiman, E., Widians, J.A., Wati, M., Hairah, U., and Ugiarto, M., Biodiversity information system: Tropical rainforest borneo and traditional knowledge ethnic of dayak. *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10. No. 1-9, 2018, pp. 59-64.
- [7] U. Hairah, A. Tejawati, E. Budiman and F. Agus, "Borneo biodiversity: Exploring endemic tree species and wood characteristics," 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, 2017, pp. 435-440. doi: 10.1109/ICSITech.2017.8257152.
- [8] Haeruddin., Johan, H., Hairah, U., and Budiman, E., "Ethnobotany database: Exploring diversity medicinal plants of dayak tribe borneo," In International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Institute of Advanced Engineering and Science, 2017, pp. 120–125. doi: 10.1109/EECSI.2017.8239094.
- [9] E. Budiman and S. N. Alam, "Database: Taxonomy of plants Nomenclature for borneo biodiversity information system," 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, 2017, pp. 1-6. doi: 10.1109/IAC.2017.8280642.
- [10] Haeruddin, H. Johan, U. Hairah and E. Budiman, "Ethnobotany database: Exploring diversity medicinal plants of Dayak tribe Borneo," 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, 2017, pp. 1-6. doi: 10.1109/EECSI.2017.8239094
- [11] E. Budiman and O. Wicaksono, "Measuring quality of service for mobile internet services," 2016 2nd International Conference on Science in Information Technology (ICSITech), Balikpapan, 2016, pp. 300-305. doi: 10.1109/ICSITech.2016.7852652
- [12] E. Budiman and S. N. Alam, "User perceptions of mobile internet services performance in borneo," 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, 2017, pp. 1-6. doi: 10.1109/IAC.2017.8280643
- [13] Sanchez D., Mendez O., Florez H.: An Approach of a Framework to Create Web Applications. In: Gervasi O. et al. (eds) *Computational Science and Its Applications – ICCSA 2018*. ICCSA 2018. Lecture Notes in Computer Science, Springer, Cham, 2018, vol 10963.
- [14] Arnowitz, J., Arent, M. and Berger, N., "Effective prototyping for software makers", Elsevier, 2010.
- [15] Zirkiewicz, A., Milosz, M., (2015). *Selecting A Php Framework For A Web Application Project – The Method And Case Study*.
- [16] International Code of Nomenclature for algae, fungi, and plant, 2018. <https://www.iapt-taxon.org/nomen/main.php>, last accessed 2018/09/15.

Acknowledgments

Thanks to the **Directorate General of Strengthening for Research and Development, Ministry of Research, Technology and Higher Education** of the Republic of Indonesia who have given "financial support" to this research. Thanks also to the Institute Research and Community Service (*Lembaga Penelitian dan Pengabdian Masyarakat: LPPM*) Mulawarman University for all the support and mentoring assistance during this research.