

The Path to an Effective ILMS: Development and Key Considerations YII 2

Roger Rosario Reyes, Rosa Gómez Domínguez, Miguel Pérez Vasconcelos, Fidelio Castillo Romero

National Technological Institute of Mexico / Technological Institute of Villahermosa

Abstract

At present, scientific and technological advances are transforming the way people live and redefining the role of public and private institutions at the local, national, and global levels. In this context of constant change, libraries emerge as crucial actors, and to efficiently manage their functions, the importance of Integrated Library Management Systems (ILMS) stands out.

This article addresses the development of an ILMS using the Yii2 framework in PHP, following the agile SCRUM methodology and employing the MariaDB database management system. The advantages include efficiency in development, organization through Yii2's modularity, outstanding performance, and a focus on security. Extensive documentation and support have facilitated understanding and maximized Yii2's capabilities in the project.

The use of Yii2 for ILMS development offers efficiency and robustness, with the constant need to adapt to the changing demands of the library environment.

Resumen

En la actualidad, los avances científicos y tecnológicos están transformando la forma en que las personas viven, y redefiniendo el papel de las instituciones públicas y privadas a nivel local, nacional y global. En este contexto de cambio constante, las bibliotecas emergen como actores cruciales, y para gestionar eficientemente sus funciones, destaca la importancia de los Sistemas Integrados de Gestión de Bibliotecas (ILMS, por sus siglas en inglés).

Este artículo aborda el desarrollo de un ILMS utilizando el Framework Yii2 en PHP, siguiendo la metodología ágil SCRUM y utilizando el sistema de gestión de base de datos MariaDB. Las ventajas incluyen eficiencia en el desarrollo, organización a través de la modularidad de Yii2, rendimiento excepcional y un enfoque en la seguridad. La extensa documentación y el soporte existente, han facilitado la comprensión y maximizado las capacidades de Yii2 en el proyecto.

El uso de Yii2 para el desarrollo de ILMS ofrece eficiencia y robustez, con la necesidad constante de adaptarse a las demandas cambiantes del entorno bibliotecario.

Keywords: Yii2, ILMS, Management, Development, Methodology

Palabras Clave: yii2, SIGB, Gestión, Desarrollo, Metodología

1. INTRODUCTION

In the current era, scientific and technological advancements are transforming people's lifestyles while inducing a shift in the roles played by local, national, or global public and private institutions. This need for adaptation arises not only from social and technological changes but also due to the rapid transformation of our world. All this information must be accessible to a constantly changing society, and it is at this point that the role of libraries becomes crucial (IFLA 2001).

As crucial as libraries are, so are their processes, particularly those that control the management of collections, users, and library staff. To achieve this supervision, the Integrated Library Management System (ILMS) emerges, with its origins dating back to the 1960s in the United States Congress. This system consists of software specifically designed with computer tools, linked to a database, encompassing all the regulatory and design elements necessary to fulfill its purpose.

ILMS provides essential functions for the operation of a library, automating everyday tasks, catalogs, and especially library customization, leading to advantages such as better decision-making and loan control.

This article addresses the process of creating an Integrated Library Management System (ILMS) using the Yii2 framework in PHP. To carry out this project, the agile SCRUM methodology and the MariaDB database management system are employed. The purpose is to provide a comprehensive view of how software of this nature is developed, especially when using this framework. Concrete examples of Yii's operation, instances of use, and generated code segments are presented to encourage a better understanding and learning of this tool.

2. INTEGRATED LIBRARY MANAGEMENT SYSTEM

The advancement of Information and Communication Technologies (ICT) has transformed the way institutions carry out their activities. In the case of libraries, process automation has been a necessary response to enhance their efficiency and effectiveness.

In the 1960s, the United States Library of Congress initiated the process of automating its bibliographic collections. This initiative was a milestone in the history of libraries, marking the beginning of the development of Integrated Library Management Systems (ILMS) (Bárceñas Rodríguez et al., 2013).

ILMS are computer tools that allow the automation of the most common activities and processes in a library. These tools are based on managing information in an inventory system, controlled by a unique and relational database.

ILMS offer a wide range of functions, including:

- Cataloging: creation and maintenance of the library catalog.
- Loans: management of material loans and returns.
- Acquisitions: management of the purchase, reception, and registration of materials.
- User Management: registration and maintenance of library user data.
- Reports: generation of reports on the library's operation.

2.1 ADVANTAGES OF USING AN ILMS

Integrated Library Management Systems (ILMS) provide significant advantages to libraries by optimizing processes through the automation of tasks such as cataloging, loans, acquisitions, and user management. This not only enhances library efficiency and effectiveness but also streamlines user access to the catalog and borrowed materials. ILMS simplify the tasks of librarians, allowing them to focus on strategic activities and manage collections more efficiently. Additionally, they prevent duplications by providing information about existing materials, thereby contributing to improving the library environment with quick and personalized service for users.

2.1.1 Evaluation of Library Needs

Before selecting an ILMS, it is important to assess the library's needs. This includes considering the type of materials it houses, taking into account special situations and various usage scenarios. For example, in a university library, books are often loaned for a longer duration to students working on theses or graduation projects compared to regular students, which is a special scenario.

It is important to consider ILMS usage scenarios, especially regarding loans. For instance, if the library offers interlibrary loans, the ILMS must be able to manage them.

2.1.2 Specific Advantages of ILMS

ILMS offer a series of specific advantages for libraries, including:

User Management: ILMS can assist libraries in managing their users, from registering new users to renewing their loans.

Reports: ILMS can help libraries generate reports on their operations, which can be useful for decision-making.
Materials Loan Management: This becomes a notable advantage due to precise control over loan and return dates. Therefore, the creation of such a system demands high accuracy and a deep understanding of the specific functioning of the library in which it is implemented.

While each library has its own methodology for acquisitions, whether through purchases, donations, or internal production, it is crucial to record all entries in the library catalog. Here is where one of the main advantages of an ILMS stands out: upon entering new material, it becomes immediately available to the user and appears in search results.

3. DEVELOPMENT OF AN ILMS

Currently, the development of an Integrated Library Management System (ILMS) is gaining great significance in the context of library automation, digitization, and virtualization. However, this task is not simple, as it involves considering a series of factors, among which stand out the appropriate methodology for the project, conducting the necessary modeling, and selecting the tool that will be used in the system development (Breeding, Marshall (2012)).

The choice of the tool is a task that requires taking into account the specific needs of the library, project requirements, availability, and the learning curve.

3.1 Methodology

Methodology is a crucial part of the development of any system, including Integrated Library Management Systems (ILMS). According to Maida, EG, Pacienza, J. (2015), methodology is not only a specific task in the project but also allows conceiving, implementing, and maintaining software products from identifying the need to achieving the goal for which it was created.

R. Pressman (2010) emphasizes that software development is not only determined by the process and methodology but also involves specific stages:

- **Communication:** Involves communicating with the client, gathering requirements, and addressing objectives.
- **Planning:** Defines the work to be done through a work or development map that describes resources, tasks, risks, probable outcomes, deliverables, and activity scheduling.
- **Modeling:** Involves creating a sketch to have an overall view of the object and better understand and solve the problem.
- **Construction:** Encompasses development activities related to coding and testing.
- **Deployment:** Involves delivering the software to the client for evaluation to ensure quality and usability assessment.

Due to the system requirements and the library's characteristics, the decision was made to employ the agile Scrum framework, known for its focus on good teamwork practices, chaos control, and specific control mechanisms. It allows partial and regular deliveries of the product through Sprints and Epics, controls the work team, enhances productivity, is flexible to changes, and aids in risk mitigation. However, a good understanding and team training are necessary for successful implementation (The Scrum Framework n.d.).

Below is the detailed workflow according to the Scrum methodology in various epics:

Epic 1: Requirements Engineering and Training

Duration: 8 weeks

Location: First sprint

Objective: Gather requirements, create business process diagrams, and define user stories using the 'as, I want, for, conditions' structure. Focus on achieving reliable requirements engineering. Yii2 technology and MariaDB are utilized. At the end, a review meeting is held with the client.

Epic 2: Data Modeling

Duration: 2 weeks

Location: Second sprint

Objective: Create the database model based on business needs and requirements engineering. Navicat Premium is used for physical design. A review with the client is conducted at the end to proceed to the next stage.

Epic 3: Interface Design

Duration: 3 weeks

Location: Third sprint

Objective: Initiate system development through screen proposals. The designer presents proposals for programming. Client authorization is required before effective programming.

Epic 4: Component Programming

Duration: 18 weeks

Location: Third sprint

Objective: Central part of development, involving component programming and initial save tests. Active client participation is emphasized. At the end of this stage, the first version of the system is presented to the client for reviews and testing.

Epic 5: Testing

Duration: 3 weeks

Location: Fourth sprint

Objective: Conduct tests before final deployment. The client experiences the system's operation for the first time. Real data provided by library personnel is used for specific tests, faithfully representing the system's functionality.

Epic 6: Deployment and Training

Duration: 2 weeks

Location: Fourth sprint

Objective: Client software registration, final deployment, and library staff training. The system is presented to the student community of the owning Institute as the final phase of the Project.

3.2 UML Modelling

Detailed planning is essential in the development of a software project, such as an Integrated Library Management System (ILMS). In this context, the Unified Modeling Language (UML) serves as a valuable modeling tool to represent the structure, behavior, and interaction of software systems. UML can be used at all stages of the software development process, from the requirements analysis phase to the implementation phase.

UML primarily consists of a language and modeling process, using a graphical notation to express designs. This notation provides guidance on the steps to take in design, fundamentally based on Object Orientation. Object Orientation in UML comprises concepts such as objects, classes, specialization, and composition, along with principles like abstraction and encapsulation. It is crucial to highlight that, according to Fowler and Scott (1999), an object in UML can be a real-world entity with physical existence (like a tangible object) or without it (like a legal text).

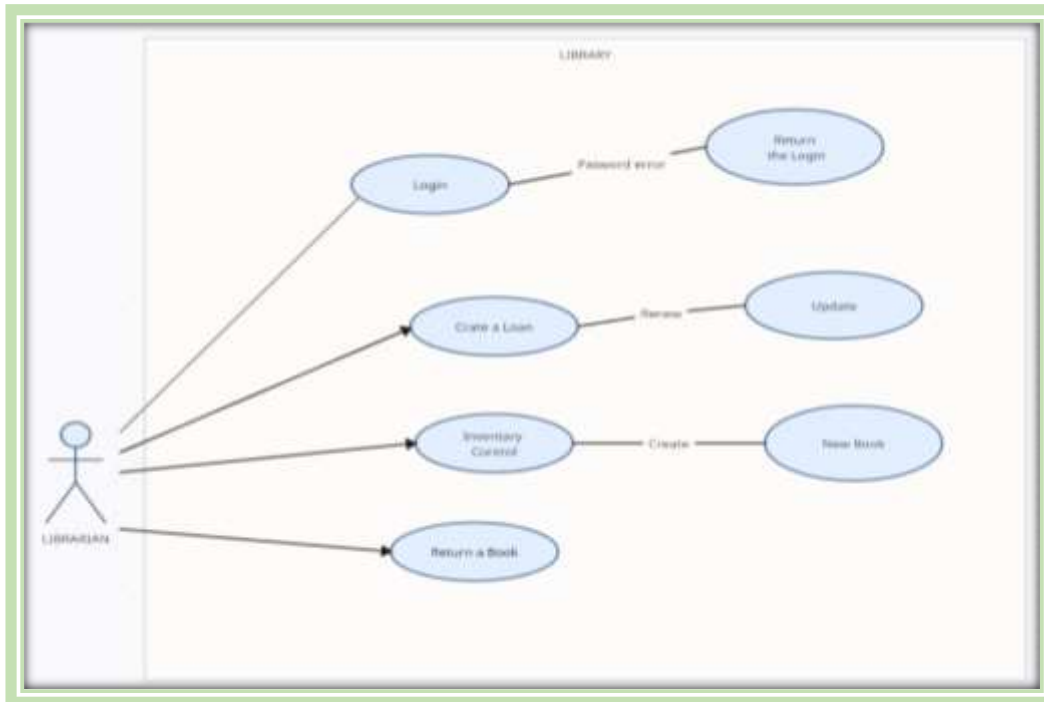


Figure 1. Use Case Diagram for the Coordinator User in an ILMS (created by the author)

- Class Diagram: Represents the classes, attributes, and methods of a software system.
- Object Diagram: Represents the objects of a software system and their relationships.
- Component Diagram: Represents the physical components of a software system and their relationships.
- Package Diagram: Represents groups of related UML elements.
- Use Case Diagram: Represents the system's use cases and the relationships between them.
- Sequence Diagram: Represents the sequence of events that occur in a use case.
- Collaboration Diagram: Represents the collaboration relationships between objects.
- Activity Diagram: Represents the flow of control and data in a system.

4. USE OF YII2 AS A DEVELOPMENT TOOL

In the progression of developing an Integrated Library Management System (ILMS), the choice of a development tool plays a crucial role. In this case, the article focuses on the use of Yii Framework 2.0.2.

4.1. What is Yii?

Yii, according to its creators, is a high-performance PHP framework based on components, designed for agile development of modern web applications. This framework offers a component-based architecture, provides extensive support and widgets (graphical elements used on websites), and follows the Model-View-Controller (MVC) pattern. Its versatility allows it to address the development of large-scale applications. Additionally, it stands out for its high performance and the elegance and simplicity in code writing, making it easy for programmers to identify code structures.

4.2 Why Yii frame work?

The choice of a development framework is not a simple task and depends on various considerations, such as the specific needs of the project and the client. Yii2 stands out as a robust option, offering performance, security, modularity, and efficiency in web development, especially in the realm of agile development.

In terms of performance, Yii2 is designed to be fast and efficient, employing advanced techniques in database queries and component-based modular structures for code management and reuse. Moreover, it provides integrated security against common web vulnerabilities. With AJAX and Bootstrap support, Yii2 makes it easy to create modern and attractive user interfaces (Development, S.E. n.d.).

Development speed is crucial in today's agile development and competitive environment. Yii2 addresses this need with Gii, a code generation tool that speeds up the process by creating initial codes for models, controllers, forms, and more (Development, S.E. n.d.).

Yii2 stands out as an agile tool with an accessible learning curve, abundant documentation, and good accessibility. These features, combined with its performance and security, make Yii2 a suitable choice for developing a modern, secure, and versatile ILMS.

5. DEVELOPMENT OF THE ILMS USING YII

In this phase, it's time to employ the selected tool. Yii can be obtained for free through the following link: <https://www.yiiframework.com/doc/api/2.0>. Once obtained, the next step is to install it in the "www" folder of the preferred web server. With the installation complete, and bearing in mind that Yii uses PHP code, we can start working using a code editor.

At this point, we experience the first significant benefit of the tool compared to traditional methods. While in conventional development, the programmer must write all the code to link the database to the system, Yii already includes this pre-written code. It only needs to be edited to establish the connection. As a modular tool, it organizes the code into various folders, simplifying the handling and control of development. These folders include: assets, commands, configuration, controllers, mail, messages, models, runtime, tests, vagrant, vendor, views, web, and widgets.

To connect to the database, access is made through the configuration folder, where the code corresponding to the database connection is located, generated automatically by Yii2 Framework.

```
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=BIBLIOTECA',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
];
```

Figure 2. Code segment extracted from the Yii2 Framework repository

After making modifications to the database connection file, Yii verifies its existence in the corresponding manager and establishes the connection. With this step completed, we can generate the other basic codes for development.

5.1 DATA BASE

A crucial aspect that must not be overlooked is that, for Yii to function as desired, it is imperative that the database is fully defined, with its tables and relationships established. This is essential as Yii needs this structure to automatically generate the code corresponding to the tables using its code generation tool, Gii.

In the Integrated Library Management System (ILMS) addressed in this article, some of the fundamental tables that should be included in such software are selected. These tables encompass user, books, author, publisher, section, loans, and fines. Yii2 will use these tables as a foundation to generate the essential codes that will facilitate the work and development of the comprehensive library management system. As illustrated in the image below, establishing relationships between these tables is crucial for Yii to understand the dependency between them.

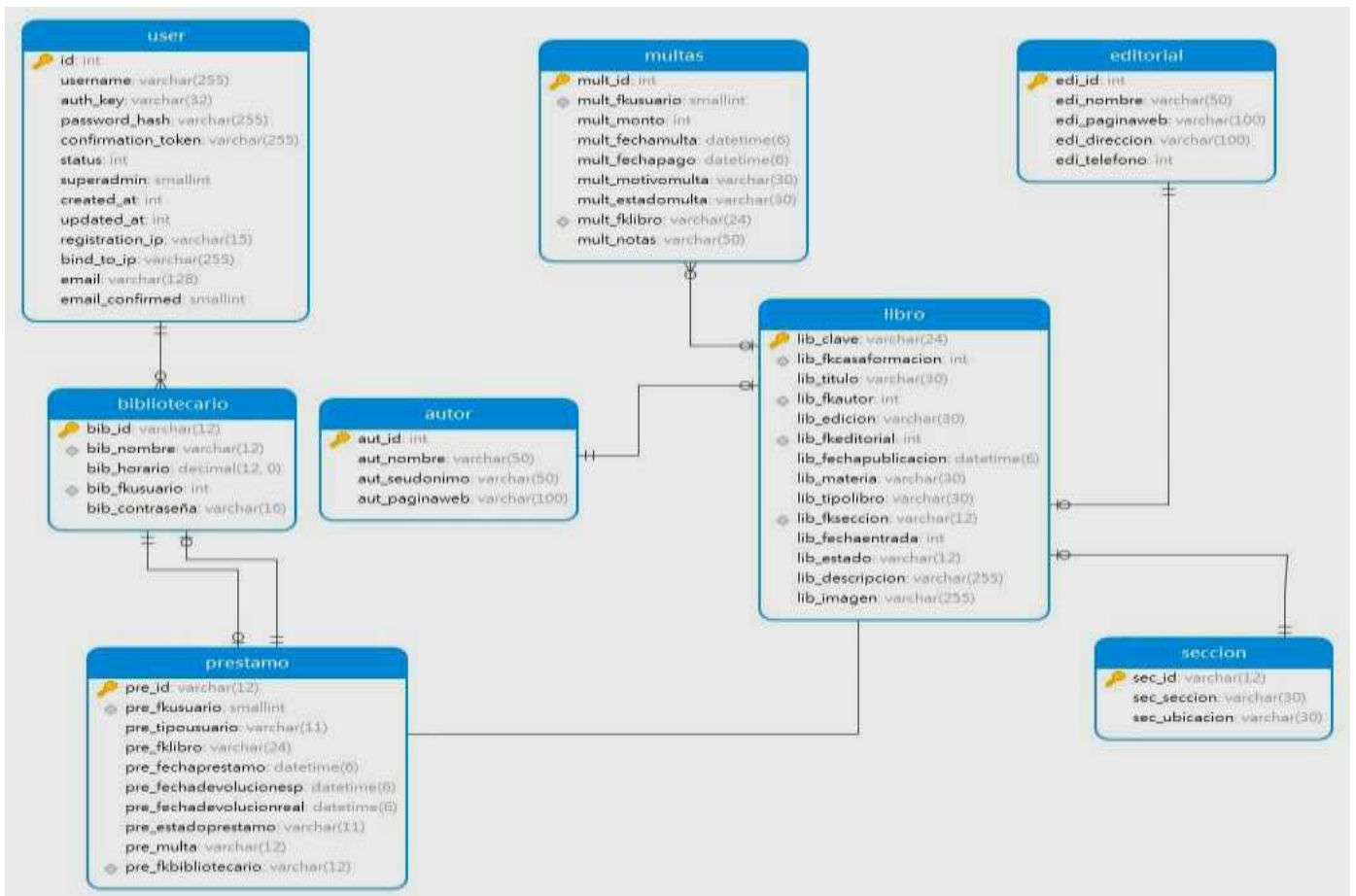


Figure 3. Tables in a database for a ILMS (created by the author)

5.2 Models

Now that the database tables are defined, it's time to generate the models, known as 'models' in Yii2. These models constitute the model layer in the Model-View-Controller (MVC) design pattern. They represent the merging of data and business logic of the application, linking directly with the tables in the database.

The creation of these models is carried out through Yii2's Gii code generator by directly connecting to the database, so it is essential that this connection is well established and connected with the database name and password to allow Yii2 to find the database and connect to read the existing tables and their relationships. Below is an example applied to the previously mentioned database.

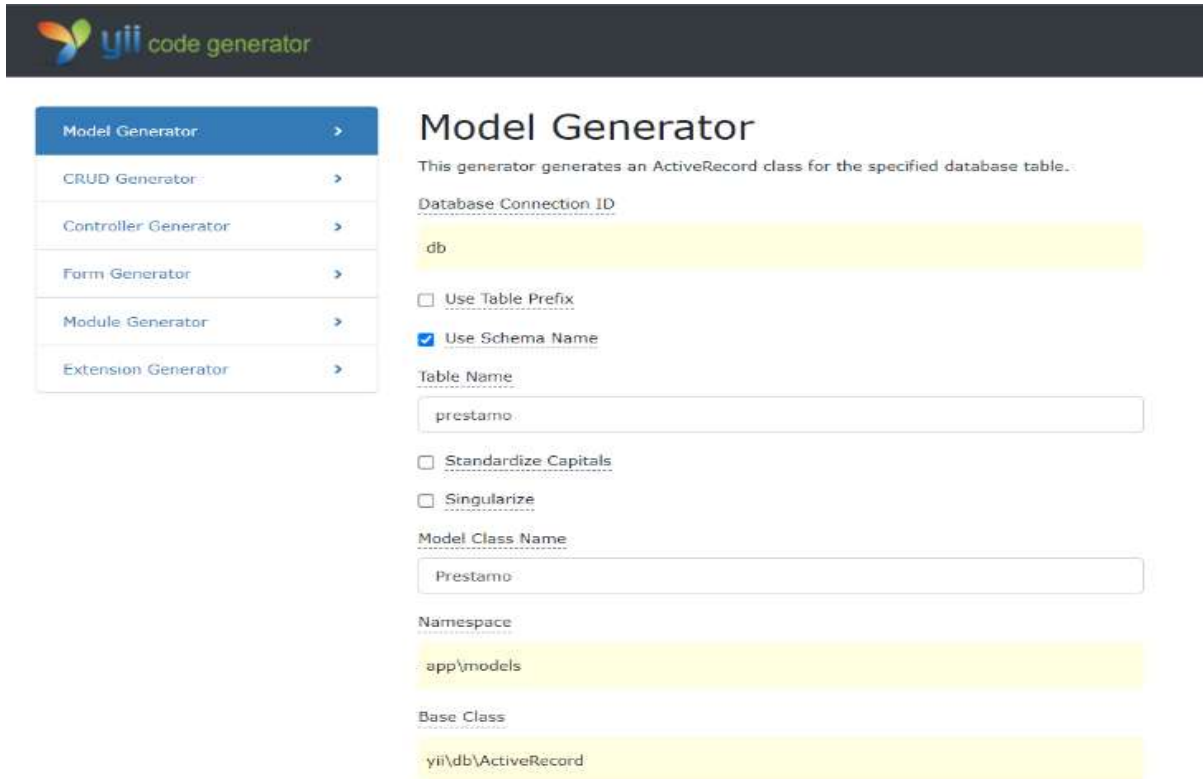


Figure 4. Screen of generating a model from the Gii generator of yii2

The figure above exemplifies the model generator. This process must be repeated for each of the tables in the database. These tables play a crucial role in the system's interaction with the database, as they are critical to data validation in that environment.

5.3 CRUDS

After creating the models, they are used to generate CRUDs, which consist of code segments responsible for the fundamental operations of any system: Create, Read, Update, and Delete. In this context, these operations are directly related to the manipulation of models and interaction with the database. These actions are carried out through web forms, where data is entered, queries are performed, and modification and storage of updates in a retrieved model are executed.

The creation of a CRUD requires that the models are already defined, as they provide the CRUD with data from the table existing in the database. Below is an example of creating a CRUD for the 'author' table in the ILMS we are developing. In this process, segments of the controller, searches, forms, creation functions, index, update, and views are generated. Subsequently, each of these segments is modified according to specific requirements to fulfill their intended functionality, including data presentation and user interface design.

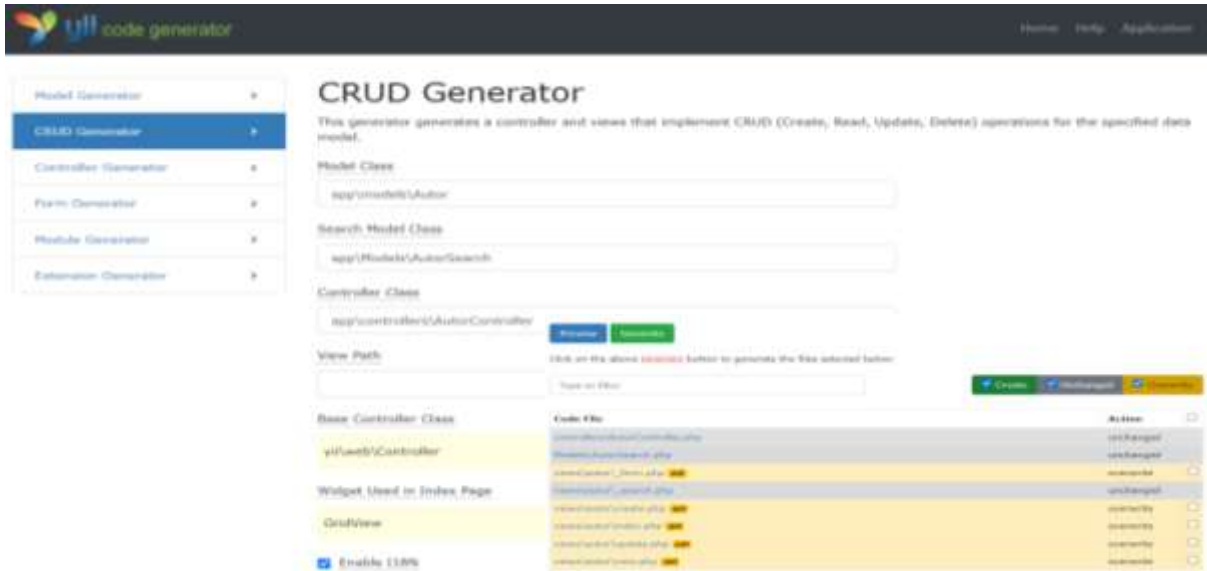


Figure 5. Screen of generating a CRUD from the Gii generator of Yii2



Figure 6. View of the authors table created by a CRUD in the Authors model for the ILMS (created by the author)

5.4 The code

After having created the CRUDs, the subsequent coding phase allows us to customize all windows, forms, and tables to meet the specific requirements of the user. Up to this point, the Integrated Library Management System (ILMS) consists of a series of independent screens, presented in a basic and unformatted manner with respect to the database. At this stage, the programmer takes on the responsibility of coding all the necessary elements for the final presentation.

We can envision Yii as an expert who constructs a solid structure for the house, complete with walls, rooms, and a roof, leaving the programmer the task of establishing security functions, decorating the environment, assigning specific functions to each room, determining the type of flooring, and applying desired finishes. Once each programming segment is created, the configuration of security measures begins. One option for this is to use webvmark, through which we can define roles and permissions using simple code, as shown below.

```

public function behaviors ()
{
return [ 'ghost-access'=> [ 'class' => 'webvimark\modules\UserManagement\components\GhostAccessControl',
],
];
}

```

This security code must be adjusted in all controllers created for each view in the system. However, this step is not the only one required; it is also essential to make modifications in layouts in key areas such as the main, login, and management of requests from newly created users.

Another section of the code that requires modification is the 'site\index' of the system. At this point, all accesses for each of the CRUDs and models created by Yii are established, becoming the main view of our Integrated Library Management System (ILMS). Additionally, elements such as Ajax, Bootstrap, and CS can be added to this section to enhance user interaction with the system.

In the system referenced in this article, a simple presentation code and a dashboard that facilitates visualization in the index are employed. The following is the code segment used for this functionality.

```

<?php
use yii\widgets\ListView;
?>
<html lang="es"><head>
  <meta charset="UTF-8">
  <style>
  </style>
</head>
<body> <header>
  <div class="jumbotron text-center">
    <h1 class="display-5"><?= Yii::t('app', 'Bienvenid@s') ?></h1>
  </div>
</header>
<aside lang="es" dir="ltr" accesskey="s" class="sidebar" id="sidebar" style="background-color: #FFFFFF; padding: 7px;">
  <ul>
    <div class="site-index">
      <div class="body-content">
        <?= ListView::widget([
          'dataProvider' => $dataProvider,
          'options' => [
            'class' => 'row row-cols-lg-4 <row-cols-md-4></row-cols-md-4> row-cols-sm-4'
          ],
          'layout' => '{items}',
          'summary' => false,
          'itemView' => '_dashboard',
        ]) ?>
      </div>
    </div> </ul>
  </aside>
</body>
</html>

```

Figure 7. Code for the ILMS Index, created by the author

6. DISCUSSION AND CONCLUSIONS

In the development of an Integrated Library Management System (ILMS) using Yii2 as the main tool, various advantages and important considerations have been observed.

Discussions

Database Requirements: It is crucial to emphasize the importance of having a fully defined database before implementing Yii2. This can pose a challenge if database requirements are not clear from the outset.

Learning Curve: Although Yii2 excels in accessibility, the learning curve should be considered to maximize its effectiveness. Full familiarity with Yii2's features and capabilities takes time, but this effort translates into long-term benefits.

Flexibility and Adaptability: The choice of Yii2 is based on its performance and efficiency for agile development. However, it is vital to continually assess its flexibility and adaptability as project requirements evolve.

In summary, the use of Yii2 in ILMS development has proven to be a solid choice, standing out for its efficiency, modularity, and performance. However, attention must be paid to the initial database planning and the learning curve to fully harness all capabilities of this development tool.

Conclusions

Efficiency in Development: Yii2 has proven to be an efficient tool for ILMS development, providing pre-written functions that streamline database connectivity and essential code generation.

Modularity and Organization: Yii2's modular structure, which organizes code into specific folders, has facilitated the management and control of system development, improving code readability and maintainability.

Performance and Security: Yii2 stands out for its efficient performance and focus on security. Advanced database query techniques and integrated security measures, such as protection against web vulnerabilities, contribute to the robustness of the ILMS.

Support and Documentation: The presence of extensive support and documentation has been crucial for understanding and fully leveraging Yii2's capabilities in ILMS development.

REFERENCES

- [1] Bárcenas Rodríguez G., Montero Laurencio R., Legrá Lobaina A. A., López Huerta M. J. y Lambert Sánchez R. 2013. Biblioteca Digital sobre la base del software libre Greenstone y el modelo 5S. En: Ciencias de la Información: vol. 44, no. 2, pp. 55-64. http://www.researchgate.net/publication/263854084_Biblioteca_Digital_sobre_la_base_del_software_libre_Greenstone_y_el_modelo_5S.
- [2] Breeding, Marshall (2012). "Current and future trends in information technologies for information units". El profesional de la información, 2011, v. 21, n. 1, pp. 9-15. Doi: <http://dx.doi.org/10.3145/epi.2012.ene.02>
- [3] Carvajal, C. M. (2018, 9 marzo). METODOLOGÍA ÁGIL CRYSTAL CLEAR. Consultado el 11 de noviembre de 2023 en <https://es.linkedin.com/pulse/metodolog%C3%ADa-%C3%A1gil-crystal-clear-cristian-mu%C3%B1oz-carvajal#:~:text=Crystal%20es%20una%20metodolog%C3%ADa%20de,a%20estar%20en%20un%20proyecto>.
- [4] Development, S.E. (s/f). Introducción: Acerca de yii. Yii Framework. Recuperado el 10 de noviembre de 2023, de <https://www.yiiframework.com/doc/guide/2.0/es/intro-yii>
- [5] Fowler, M., & Scott, K. (1999). UML gota a gota. México: Pearson Educación.

- [6] Maida, EG, Pacienza, J. Metodologías de desarrollo de software [en línea]. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería “Fray Rogelio Bacon”. Universidad Católica Argentina, 2015. Disponible en: <https://repositorio.uca.edu.ar/handle/123456789/522>
- [7] Roger, S. Pressman, (2010) ingeniería del software. Un enfoque práctico (7ma Ed.), Mc. Graw Hill
- [8] The Scrum Framework poster. (s. f.). Scrum.org. consultado el 10 de noviembre de 2023 <https://www.scrum.org/resources/scrum-framework-poster>
- [9] IFLA (2001) Directrices IFLA/UNESCO para el desarrollo del servicio de bibliotecas públicas, consultado el 27 de noviembre de 2023 en https://unesdoc.unesco.org/ark:/48223/pf0000124654_spa
- [10] Zurita Sánchez J. M. 2010. El potencial del software libre en la gestión de información en bibliotecas. En: Em Questão, vol. 16, no. 1, pp. 95-111. E-ISSN 1808-5245. http://eprints.rclis.org/14809/1/el_potencial_del_software_libre_en_la_gestion_de_informacion_en_bibliotecas.pdf

Email of the corresponding author: m22301150@villahermosa.tecnm.mx; rosa.gd@villahermosa.tecnm.mx