EMERGENT: JOURNAL OF EDUCATIONAL DISCOVERIES AND LIFELONG LEARNING (EJEDL)
ISSN: 2776-0995 Volume 4, Issue 2, Feb., 2023

IDENTIFYING DIFFERENT CONTENT OF USERS IN INFORMATION SYSTEMS

N. Sh. Tojiyev
Teacher, Samarkand Institute of Economics and Services

Abstract

(Matthew 24:14; 28:19, 20) Today, the extent to which this system is protected plays an important role in the development of information systems and the development of its project. The protection of information systems is significant not only by how strongly the authentication and identification system is installed, but also by the extent to which users' rights are limited.

A feature of modern information systems is the need to combine rich content with advanced technological changes. This is because, together with new information and documents, there may be systematic and semantic connections to specific information processing methods, document security, and other components of the information system. The database of such systems contains different categories of data, and most of this information may be in a hidden state for different types of users. Therefore, creating a different user's content will be an important factor, depending on the categorization of information in information systems and the distribution of tasks.

It provides opportunities such as creating different user content, deploying tasks in very large-scale information systems, and protecting hidden information. These opportunities are done by limiting or expanding user rights.

In the case of the Yii PHP Framework, which has become popular in the development of web applications in recent years, we will look at the distribution of roles to users. It is known that the Yii PHP Framework is a fast-paced PHP Framework designed to quickly create modern web applications. It can be applied to create all kinds of applications. Yii only

relies on the philosophy of writing simple and attractive code without complicating the design by following a certain mold of design.

Below we talk about RBAC, one of the systems of the Yii PHP Framework that is widely used to divide users into roles.

RBAC (Role Based Access Control)-Yii is a role-based access system in the PHP Framework that is centralized and uses the simplest methods to manage access to the information system. Its main advantage is that the authorization hierarchy can be used in a flexible way without changing the code of the entire information system. This system is considered preferable to ACF access, divorcing the users into guest and registered user categories. We may allow or ban certain actions of the user in the information space, depending on the category.

The RBAC also divides authorized users into groups, allowing each of them to grant separate rights. The names and numbers of such groups are unlimited. Yiida's system has three main joints:

- Rollar (Role)
- Vazifalar (Task)
- Operations (Operaton)

Hierarchy of roles. The most important and necessary thing in the RBAC system is the hierarchy of elements. Through this hierarchy, we can find out how flexible the roles in the system are and the control code needs to be changed. Here's a look at each authorization element:

Operations are the lowest element of authorization.

Roles are the highest element of authorization, in which groups and their functions are displayed.

Tasks are a non-binding element between an operation and a role that allows us to reduce their practical rights using BizRule. Simply put, it can be equated with the concept of a filter.

BizRule PHP kodi b o'lib, u⁄oneor qoidani mua yyan foydalanuv c higa Do you need a handash?‖ should give a clear answer to the question. The chart below (Fig. 1) shows the usual hierarchy, in which the control code checks transactions and the roles are attached to users. However, Yii does not prevent us from checking other items in the control system, such as user role. But it is important to remember that this can lead to a loss of centralized management advantages.

Information available in the information system can usually be deleted (delete), created (created), readed, and updated.

The first thing you should pay attention to when designing an RBAC system is to take into account all possible transactions.

Checking the above-mentioned delete, create, read, update operations is provided in the following application code:

```
if(Yii::app()->user->checkAccess('create'))
{
create
}
if(Yii::app()->user->checkAccess('update'))
{
// Yangilash
}
```

Once we have developed the operations, we will be able to move on to roles. From existing operations, we can separate the following roles: Reader, Manager, Author.

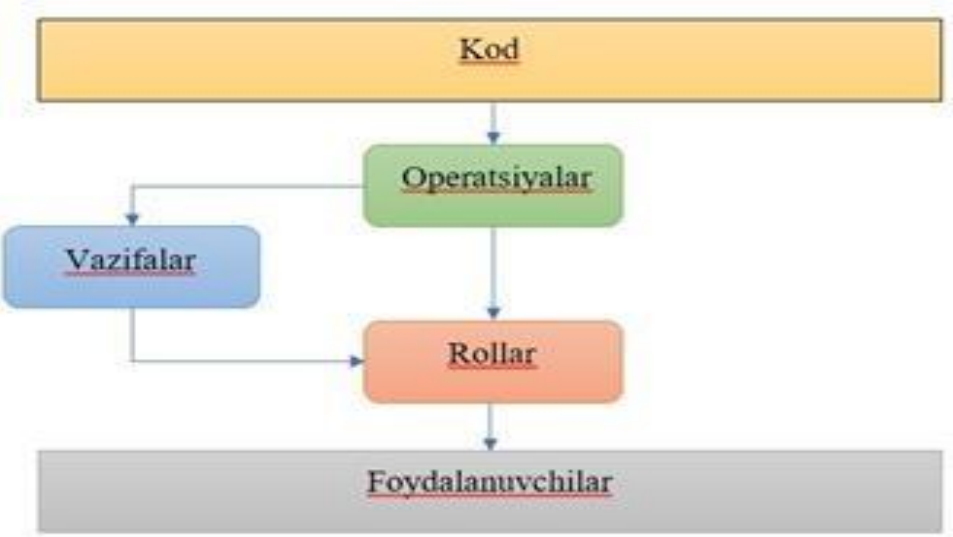

Figure 1. Hierarchy of elements.

The hierarchy of elements is as follows:

- Reader
- read; Read-only
- Author
- Read //O‘qish
- Create //Yaratish
- Manager
- Read //O‘qish
- Create //Yaratish
- Delete //O‘chirish
- Update //Yangilash

These roles can be attached to specific users. However, it is better to create another abstract that further summarizes roles and attach it to users, such as:

- guest
- Reader
- authorized
- Author
- presenter
- Manager

In conclusion, if the Yiida role mechanism is used correctly, it will be possible to create a truly flexible information system. The quest to create or adapt GUI solutions to manage roles leads to the system's head going into the hard street, with an increase in nonsense writing.

That is why the revised style is one of the most effective ways to divide roles into categories in Yii projects.