**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE**
**SUMY STATE UNIVERSITY**
**UKRAINIAN FEDERATION OF INFORMATICS**

# PROCEEDINGS
OF THE V INTERNATIONAL SCIENTIFIC CONFERENCE
# ADVANCED INFORMATION SYSTEMS AND TECHNOLOGIES

## AIST-2017
**(Sumy, May 17–19, 2017)**

**SUMY**
**SUMY STATE UNIVERSITY**
**2017**

# Application for System Administrator of the Mobile Game

Ye. Pysmennyi, student; Yu. Parfenenko, seniour lecturer
Sumy State University
yuliya.parfenenko@gmail.com

***Abstract* – This article deals with the development of server side architecture of administrator of web-based game application. A web-based application with implemented features for management of server side of administrator was built.**

***Key words* – web-based application, game, system administrator, client, server, data flow, control.**

## I.INTRODUCTION

Web-based game applications consist of a client side and server side. The client side is responsible for a user interface and game server connectivity, and the server side describes the application logic and represents powerful features of a quick and secure user interaction with each other, the involvement of local and "cloud" medium for storage of required data etc.

Such web-based game applications are characterized by hierarchical user role distribution, different levels and tools of run through the game, the physics of moving and placing of objects with textures, units of remuneration, mission, language etc.

There is a need to administer a data flow of the game application. That is, to be able to control dynamic elements of the game and control the correctness of the server features, without the intervention of a web-application code or directly in a database.

The project aim is to develop the server side of administrator of web-based game application with the ability to control the data flow between client and server on the website.

The server side of administrator will be able to test the existing or new features of the game, to implement CRUD – control of roles, news and languages of the game as well as to control XML and JSON data formats. This project has a functional value, since its capabilities provide the performance of tasks to control the game by administrator.

## II. A ANALYSIS OF DEVELOPMENT PROBLEM OF ADMINISTATIVE SIDE OF WEB-BASED GAME APPLICATIONS

Today, the majority of high-loaded web-based game applications use special data flow control systems of own servers for administrators. But such server sides serve only for a few responsible people who administer the game. That’s why they are locked for use by all other users. Such systems are developed for specific tasks and there are no clear common rules that must be met by created features.

Distribution of operation of high-loaded web-based game applications are an integral part when planning and implementing the architecture of their operation. As this criterion has influence on:

- high-speed-response;
- security of user interaction;
- usage the local and "cloud" medium for storage of data;
- performance;
- usability;
- user interface etc.

Once the web-based game application is downloaded by a client (or is opened in a browser), the client begins to interact with its interface. Thus, any dynamic changes are sent to the server on which the received data are processed and after this the processed data are sent back to the client. In turn, the user gets the result in their device or browser.

To encourage the users to use the web-based game applications, scripts and storylines are created, various types of user roles, levels and tools of run through the game, motion trajectory and location of objects with textures, units of remuneration, mission are developed. All these attributes are dynamic attributes for each user. Their logic is described on the server and data about them are stored in a database. Therefore, the possibility to administer these data flows is very important and relevant task.

A similar problem occurs in the IOS strategy game "Battle of Ages". It is a mobile game application. A type of gameplay is similar to a real-time strategy with the expectation of a large number of users.

The game has a server that provides its logic written in PHP, the main database located on Amazon RDS and server for IOS - developers. The gameplay is based on the control of their own game objects for the development of a city, collection of achievements, passing missions, conducting of wars.

The main purpose of a player is to run through the game scene by forming a military and economic power and making strategic decisions on the development of the plot.

For this game the new features are gradually added. These features should be checked for correctness of operation and should able to operate with data formats such as XML and JSON [1].

It is necessary to develop the mobile game administrator application which results in absence of necessity to interfere in web-based application code [2].

This admin panel is a special management system of data flow between the client and the server.

This will allow making the process of manipulation of the game faster and more efficient by CRUD-control and testing the existing or new features of the game, if necessary.

## III. THE SELECTION OF METHODS FOR DEVELOPMENT OF MOBILE GAME ADMINISTRATOR APPLICATION

For development of server side of administrator of web-based game application a programming language and database are used; this database can quickly handle requests and maintain high load. Of the available options the most optimal solution is PHP and database MySQL.

The main methods of implementation are structured programming and object-oriented programming [3].

When creating the medium sized applications the structured programming is used, the idea of which is that the structure of the program should reflect the structure of the original problem in order to make a solution algorithm clearly evident from the text.

This approach is convenient because it allows a person to think on the object level, not use specific operators and variables. However, this method has a drawback - in structured programming any modification leads to changes in the logic of the program. Thus, the development of server side of web-based game application administrator can be not user friendly and can take a lot of time.

Object-oriented programming (OOP) is the method of program creation, which is based on the concept of the class as a structure, which describes a set of similar real-world objects and their behavior. The task which is solved by the technique of OOP is described by terms of classes and operations performed on objects of that class.

The program in this approach is a set of instantiated objects and the relationships between them. The data compared with the processes are more stable and relatively rare are changed by the system part. Hence the main advantage of the object-oriented approach is: object-oriented systems are easier to change, because their construction is based on the stable forms. This enables the system to develop gradually and does not lead to its complete rebuilt even if there are significant changes in output requirements.

OOP is widely used for frameworks that facilitate the development of web-based applications.

The framework YII2 is used for this project. Its main advantages are:

- integrated generator queries to database;
- ActiveRecord for relational and NoSQL database;
- RESTful API;
- multilevel caching support;
- easily extensible for new features;
- powerful RBAC system;
- uses standard methods of problem solving that reduces or eliminates the complexity of the code;
- integrated validation;
- facilitates teamwork by means of compliance with standards (PSR-4);
- facilitates support code using general architecture and methods;
- GII generation module of models, controllers and modules.

It is based on MVC pattern [4].

## IV. ARCHITECTURE OF ADMINISTRATOR APPLICATION OF MOBILE GAME

Server side of administrator is implemented using MVC pattern. This pattern is appropriate for creation complex projects and has become a popular strategy for building web-based application. MVC separates user interface, data and processing of user actions into three components: Model, View and Controller [5].

Model handles the state of the application.View is the representation of the user interface. Controller receives user requests, and translates them into actions that the Model should take. Then it selects the appropriate View to handle the response [6].

MVC pattern represents the possibility to be integrated with additional configurations, modules etc. [7]. This separation gives an advantage that the changes in a separate component of the system will not affect the entire logic of the server side of the game.

Game server already uses MySQL. MySQL database contains information on players, cities, weapons, game levels, number of awards and tools. Also, it stores the data of role hierarchy to enter the system being developed [8].

Component View is responsible for displaying of client side of administrator. That is, it describes a user interface that is accessible through a web-browser on website.

Controller accepts requests (depending on the actions of the administrator) on the server. This component is responsible for the logic distribution between the data processing in the Model and display in the View.

Model receives data from the Controller. It is in this component the features of CRUD-control (Create, Read, Delete, Update) of news, roles and languages of games, the control of data in XML and JSON formats, distribution of user roles is implemented. Model has a direct connection with MySQL database and connects GII module and RBAC system for role control.

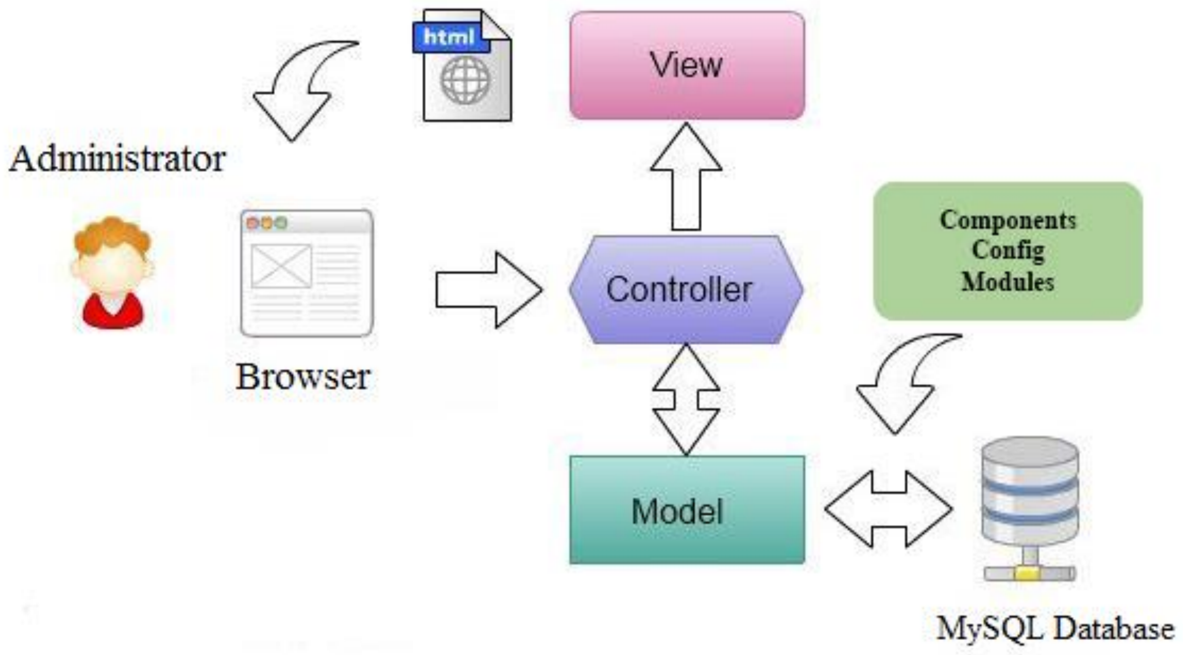

Figure 1 – System Architecture

To enter the system, you have to authorize, after which the features of server side of administrator will be available.

The administrator will be able to do the following:

- load and parse the file of XML format on the server
- convert the files from XML to JSON
- load JSON data into database tables
- appoint a new administrator of the game
- perform CRUD - control of news and languages of the games
- conduct a test checking of the existing or new features of the game if necessary
- set an access control for individual users of system being developed.

The role distribution is provided by RBAC - system (Role - based access control), which is installed as a separate extension into the server side [9].

System implements a general hierarchical RBAC interface for user, following the NIST RBAC model. It provides the RBAC functionality through the AuthManager application component. A role represents a collection of permissions. To check if a user has a specified permission, we may check if the user is assigned with a role that contains that permission.

Using RBAC involves two parts of work. The first part is to build up the RBAC authorization data, and the second part is to use the authorization data to perform access check in places where it is needed.

The administrator has the ability to control access to the system, assign new users and give them individual permissions of server side of administrator panel.

## CONCLUSION

The created architecture of server side of administrator of web-based game application will save money and time by flexible and atomic work of interconnected features. This system enables the dynamic control of elements of the game and control the correctness of features of the server without the intervention of a web-application code

## REFERENCES

[1] Poleznyie servisyi dlya veb-razrabotchika [Electronic resource]- Access mode: http://zinchenko.us/article/ru-useful-services-for-developers/

[2] Veyl, Estel HTML5. Razrabotka prilozheniy dlya mobilnyih ustroystv / Estel Veyl. - M.: Piter, 2014. – S. 661.

[3] MySQL php connect [Electronic resource]- Access mode: http://webcheatsheet.com/php/connect_mysql_database.php

[4] YII2 project basic template [Electronic resource]- Access mode: https://github.com/yiisoft/yii2-app-basic

[5] Patternyi MVC [Electronic resource] – Access mode: https://habrahabr.ru/post/215605

[6] The Model View Controller pattern in web applications [Electronic resource] – Access mode: https://zeekat.nl/articles/mvc-for-the-web.html

[7] Realizatsiya MVC [Electronic resource] - Access mode: https://www.youtube.com/watch?v=ba3M3_Myrqg&list=PLSdH7dYnlGYgQ6ElbHRpG2zRXbkgkO3zQ

[8] Velling, L. Razrabotka veb-prilozheniy s pomoschyu PHP i MySQL / L. Velling. - M.: Dialektika / Vilyams, 2016. – S. 794

[9] Web role control [Electronic resource]- Access mode: http://www.yiiframework.com/doc-2.0/guide-security-authorization.html#rbac