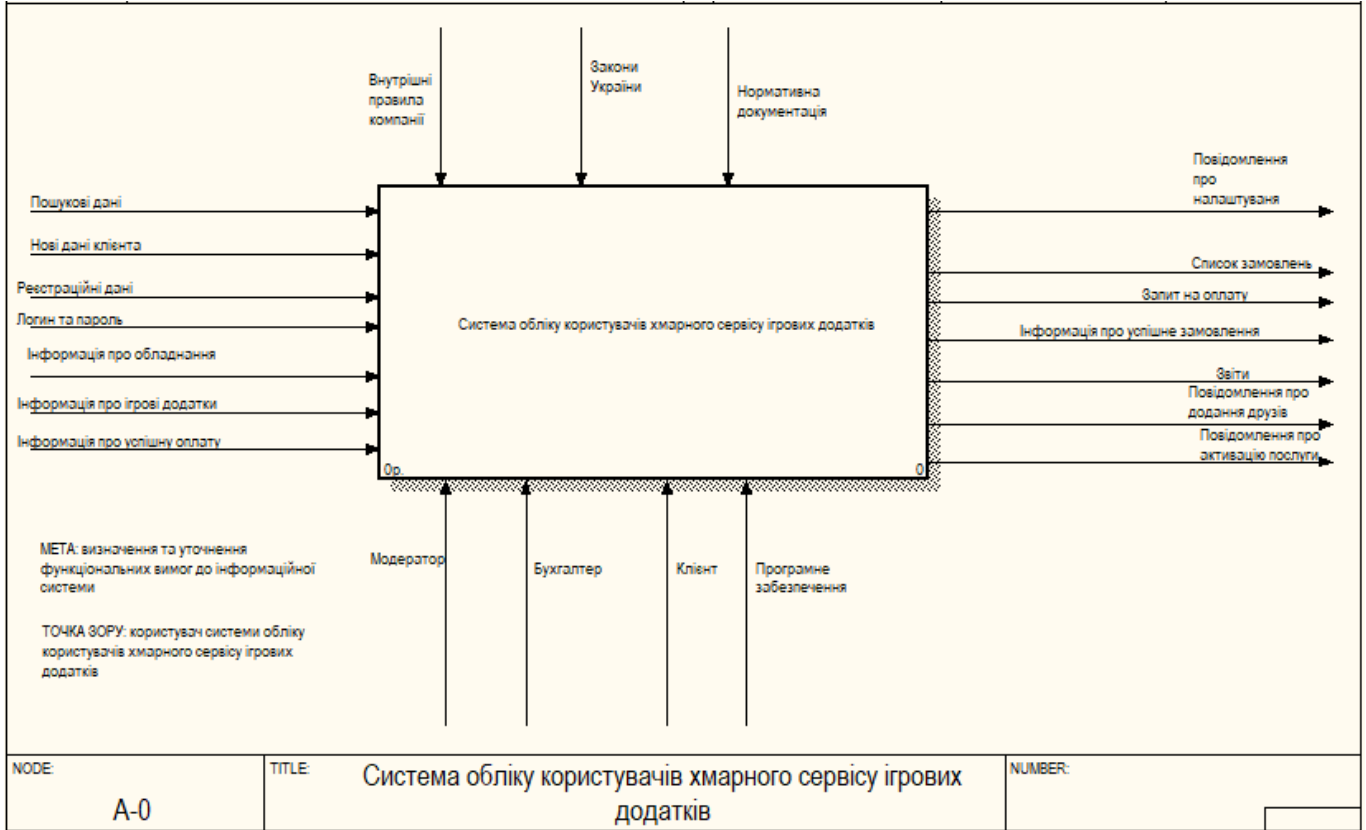


ДОДАТОК А  
Графічні матеріали

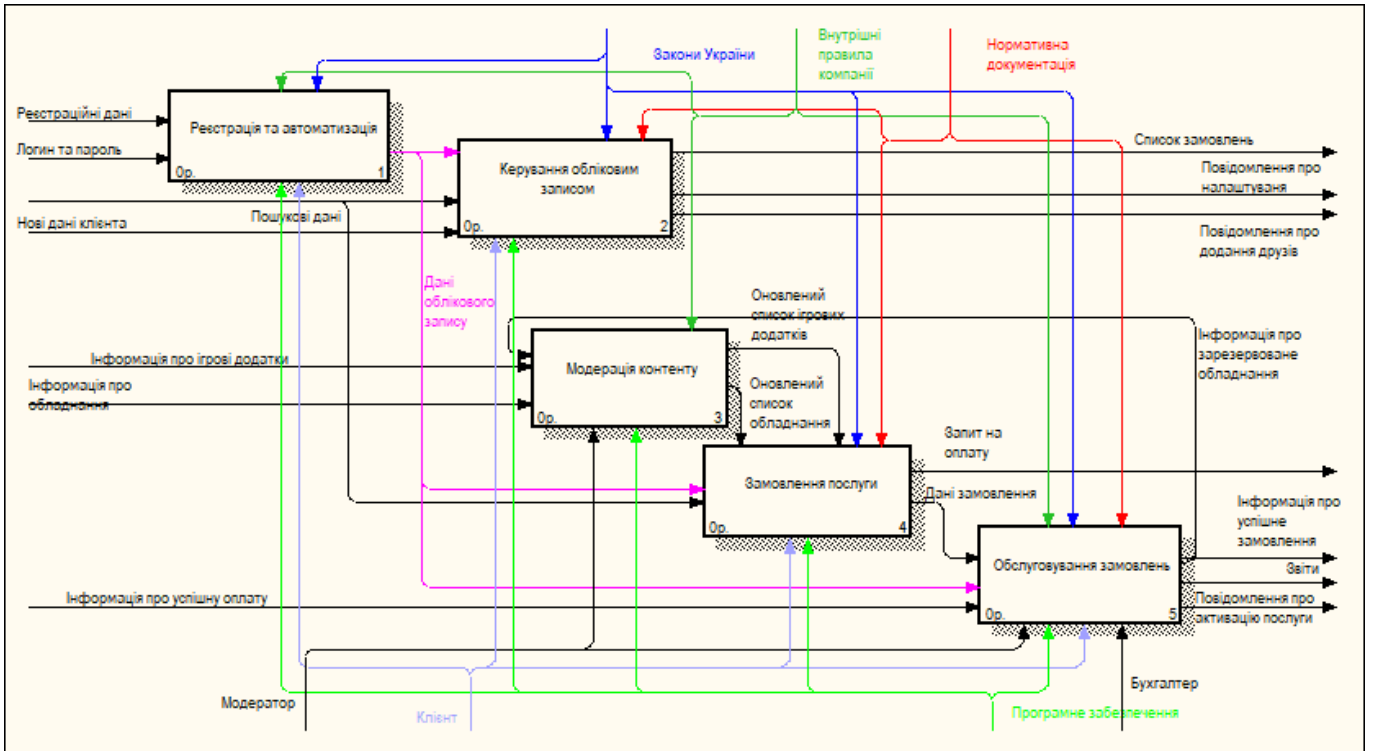
ГЮИК.502528.005



					<b>ГЮИК.502528.005</b>					
					Концептуальна діаграма системи		Лім.		Маса	М а с и т а б
Змін.	Лист	№ докум.		Підпис			Дата			
Розроб.		Муренченко П.Є.								
Перевір.		Вишняк М.Ю.								
Т. Контр.							Лист 1		Листі	

Реценз.					в 1
Н. Контр.					ХНУРЕ Кафедра СТ
Затверд.	Гребеннік І.В.				

ГЮИК.502528.005

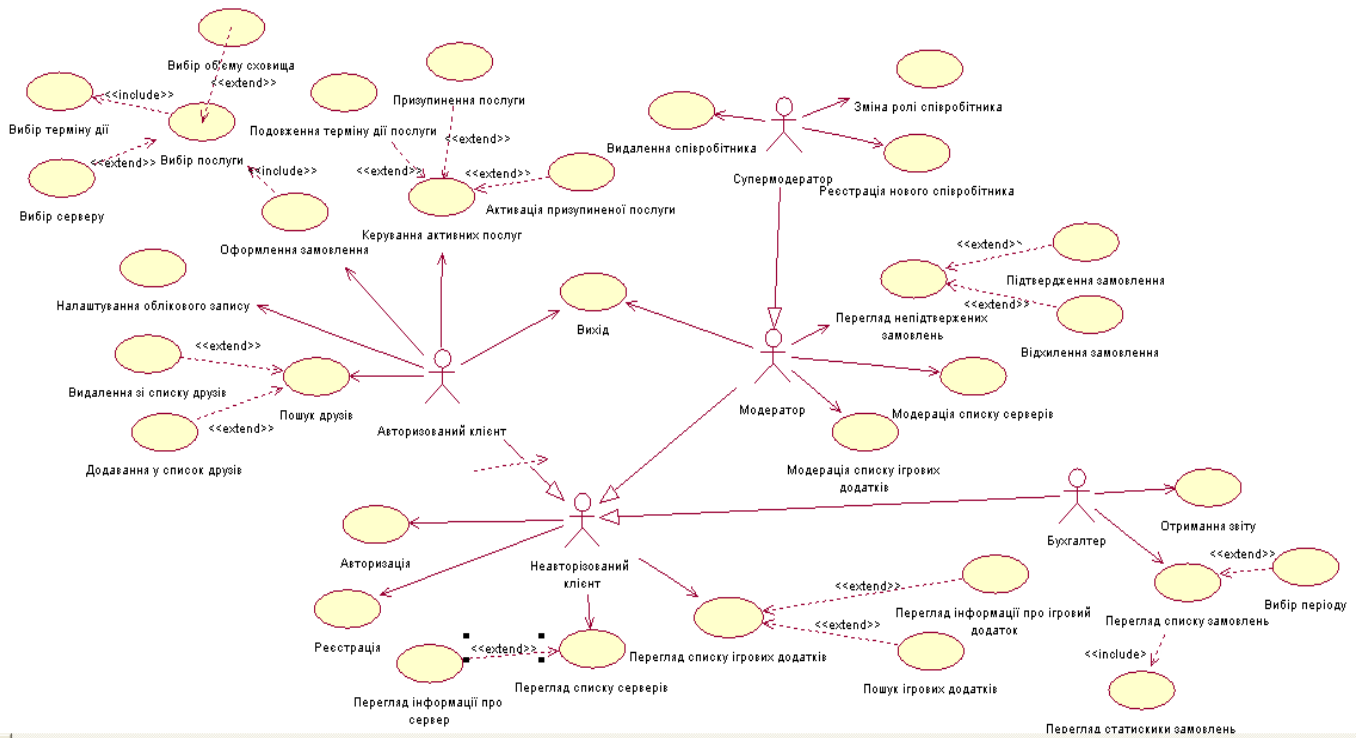


NODE:	TITLE:	NUMBER:
A0	Система обліку користувачів хмарного сервісу ігрових додатків	

					ГЮИК.502528.005				
					Діаграма декомпозиції головної функції		Лім.	Маса	М а с а б

Змін	Лист	№ докум.	Підпис	Дата	СИСТЕМИ				
Розроб.		Муренченко П.Є.							
Перевір.		Вишняк М.Ю.							
Т. Контр.							Лист 1	Листі в 1	
Реценз.							ХНУРЭ Кафедра СТ		
Н.Контр.									
Затверд..		Гребеннік І.В.							

ГЮИК.502528.005



ГЮИК.502528.005

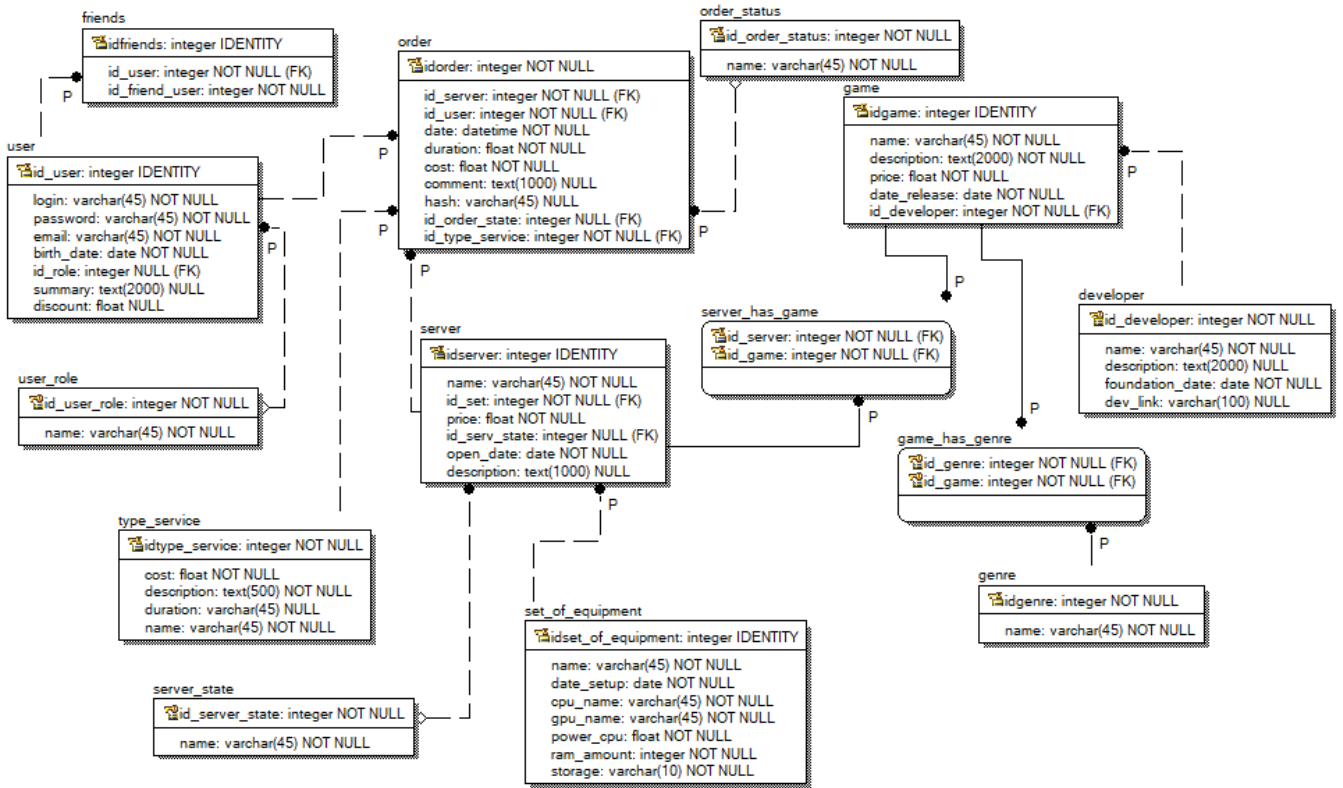
					використання			ас и т а б
Змін	Лист	№ докум.	Підпис	Дата				
Розроб.		Муренченко П.С.						
Перевір.		Вишняк М.Ю.						
Т. Контр.						Лист 1	Листів 1	
Реценз.						ХНУРЭ Кафедра СТ		
Н.Контр.								
Затверд..		Гребеннік І.В.						

ЛЮИК.502528.005

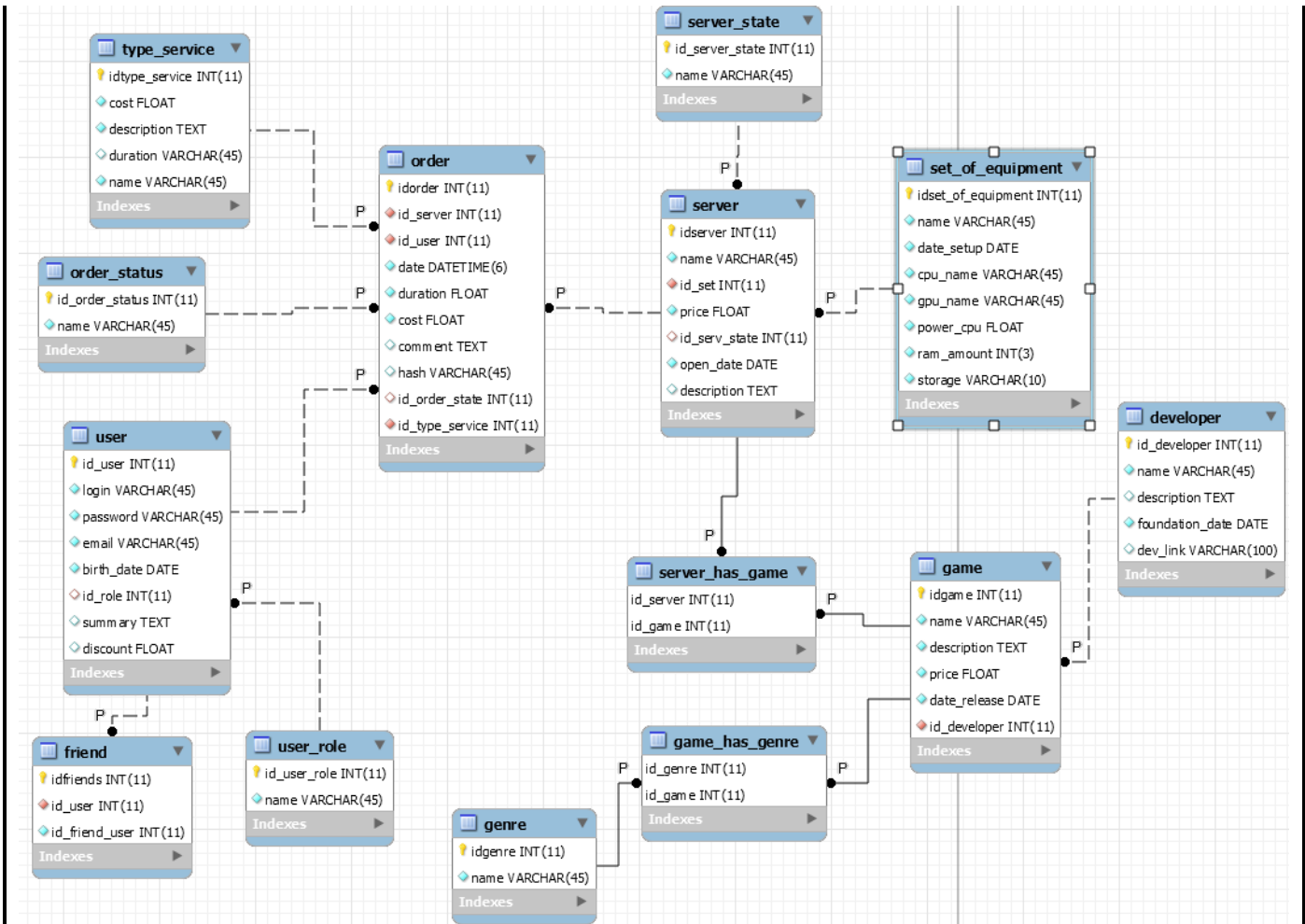


					<b>ГЮИК.502528.005</b>				
					<b>Зовнішнє середовище системи обліку хмарного сервісу</b>	Лім.		Маса	Масштаб
Змін	Лист	№ докум.	Підпис	Дата					
Розроб.		Муренченко П.Є.							
Перевір.		Вшнях М.Ю.							
Т. Контр.						Лист 1		Листів 1	
Реценз.					<b>ХНУРЭ Кафедра СТ</b>				
Н.Контр.									
Затверд..									

**ГЮИК.502528.005**



					<i>ГЮИК.502528 005</i>					
					Фізична модель бази даних системи обліку	<i>Лім.</i>		<i>Маса</i>		<i>М а с и т а б</i>
<i>Змін</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>						
<i>Розроб.</i>		<i>Муренченко П.Є.</i>								
<i>Перевір.</i>		<i>Вишняк М.Ю.</i>								
<i>Т. Контр.</i>						<i>Лист 1</i>		<i>Листів 1</i>		
<i>Реценз.</i>						<i>ХНУРЭ Кафедра СТ</i>				
<i>Н.Контр.</i>										
<i>Затверд..</i>		<i>Гребеннік І.В.</i>								
<i>ГЮИК.502528.005</i>										

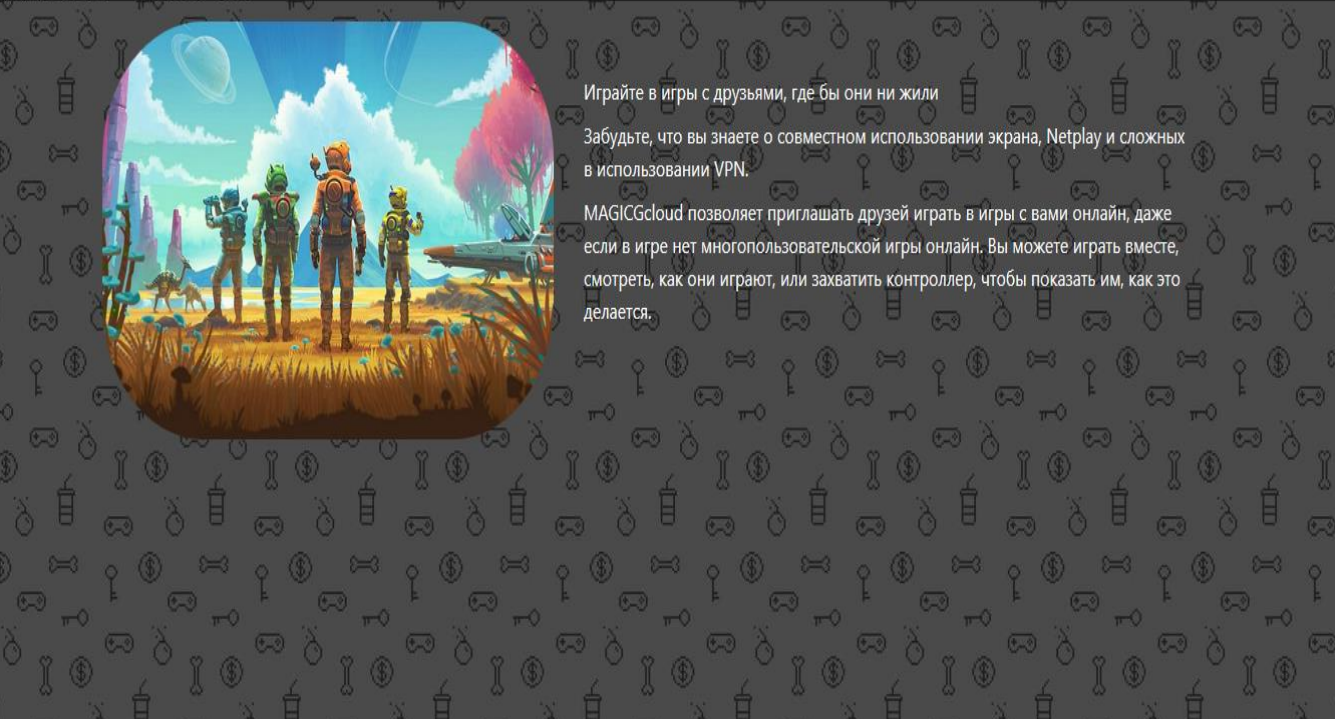


					<i>ГЮИК.502528.005</i>				
					Схема бази даних системи		<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Змін</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>					
<i>Розроб.</i>		<i>Муренченко П.Є.</i>							
<i>Перевір.</i>		<i>Вишняк М.Ю.</i>							
<i>Т. Контр.</i>					<i>Лист 1</i>		<i>Листів 1</i>		
<i>Реценз.</i>					<i>ХНУРЭ Кафедра СТ</i>				
<i>Н.Контр.</i>									
<i>Затверд..</i>		<i>Гребеннік І.В.</i>							
<i>ГЮИК.502528.005</i>									



ЛЮИК.502528.005

**MAGICG CLOUD** Главная Маркет ПО FAQ Зарегистрироваться Войти



Играйте в игры с друзьями, где бы они ни жили  
 Забудьте, что вы знаете о совместном использовании экрана, Netplay и сложных в использовании VPN.  
 MAGICGcloud позволяет приглашать друзей играть в игры с вами онлайн, даже если в игре нет многопользовательской игры онлайн. Вы можете играть вместе, смотреть, как они играют, или захватить контроллер, чтобы показать им, как это делается.

Copyright © 2019. MagicGCloud - облачный игровой сервис. [Наверх](#)

					<b>ЛЮИК.502528.005</b>				
					<b>Экранна форма головної сторінки web-системи</b>	<i>Лім.</i>		<i>Маса</i>	<i>Мас шта б</i>
<i>Змін</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>					
<i>Розроб.</i>		<i>Муренченко П.Є.</i>							
<i>Перевір.</i>		<i>Вишняк М.Ю.</i>							
<i>Т. Контр.</i>						<i>Лист 1</i>		<i>Листів 1</i>	
<i>Реценз.</i>					<i>ХНУРЭ</i>				

Н.Контр.					Кафедра СТ
Затверд..	Гребеннік І.В.				
ГЮОИК.502528.005					

The screenshot shows the MAGIC G CLOUD website interface. At the top, there is a navigation bar with links for 'Главная', 'Маркет', 'ПО', 'Заказы', 'Laress', and 'Выйти'. The main content area is titled 'Услуги' (Services) and features three service cards: 'Аренда сервера' (Server rental), 'Облачное хранилище' (Cloud storage), and 'Виртуальная сеть' (Virtual network). Below the services, there is a section for 'Игры' (Games) with a search bar and buttons for 'Найти' (Find) and 'Добавить игру' (Add game). A featured game card for 'Portal 2' is visible, with buttons for 'Подробнее' (More details), 'Изменить' (Change), and 'Удалить' (Delete).

					ГЮОИК.502528.005			
					Экранна форма сторінки модератора	Лім.	Маса	Масштаб
Змін	Лист	№ докум.	Підпис	Дата				
Розроб.	Муренченко П.С.							
Перевір.	Вишняк М.Ю.							
Т. Контр.						Лист 1	Листів 1	
Реценз.					ХНУРЭ			

<i>Н.Контр.</i>					<i>Кафедра СТ</i>
<i>Затверд..</i>	<i>Гребеннік І.В.</i>				

### ДОДАТОК Б

Перевірка тексту пояснювальної записки електронною системою на плагіат

ГЮИК.502528.012 34 01

**ЗАЯВА**

щодо самостійності виконання письмової роботи

Я, \_\_\_\_\_ Муренченко Павло Євгенович \_\_\_\_\_

(прізвище, ім'я, по батькові)

посада \_\_\_\_\_ магістр групи СПРМ-19-2 \_\_\_\_\_

кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_

заявляю: моя письмова робота на тему «Системний підхід до автоматизації обліку хмарного сервісу» \_\_\_\_\_

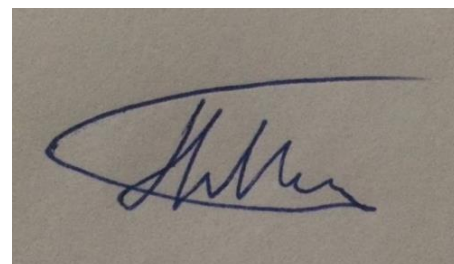
(назва роботи)

Представлена у екзаменаційну комісію \_\_\_\_\_

(спеціалізовану вчену раду, екзаменаційну комісію тощо)

Для публічного захисту, виконана самостійно і в ній не міститься елементів плагіату. Всі запозичені з друкованих та електронних джерел, а також із раніше виконаних дослідницьких робіт та захищених кандидатських і докторських дисертацій мають відповідні посилання.

Я Ознайомлений (а) з діючим положенням «Про протидію плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску письмової роботи до захисту та застосування дисциплінарних заходів.



Дата 22.05.2021 р.

Підпис



Имя пользователя:  
Кісель Тетяна Іванівна каф. СТ

ID проверки:  
1007980193

Дата проверки:  
23.05.2021 12:38:48 EEST

Тип проверки:  
Doc vs Internet + Library

Дата отчета:  
23.05.2021 12:42:38 EEST

ID пользователя:  
94940

Название файла: 2021\_М\_СТ\_СПРМ-19-2\_Муренченко\_П\_Є

Количество страниц: 60 Количество слов: 11270 Количество символов: 88418 Размер файла: 1.52 MB ID файла: 1008072640

Обнаружены модификации текста (могут влиять на процент совпадений)

## 3.22% Совпадения

Наибольшее совпадение: 0.79% с источником из Библиотеки (ID файла: 1005776259)

1.4% Источники из Интернета 70 ..... Страница 62

1.98% Источники из Библиотеки 43 ..... Страница 62

## 0.78% Цитат

Цитаты 2 ..... Страница 63

Исключение списка библиографических ссылок выключено

## 0% Исключений

Нет исключенных источников

## Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 34

Подозрительное форматирование 20 страниц

ДОДАТОК В  
Текст програми

ГЮИК.502528.005 34 01- 01 12 01 – ЛЗ

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник атестаційної роботи

\_\_\_\_\_ доц. Вишняк М.Ю.  
(підпис)

СИСТЕМНИЙ ПІДХІД ДО АВТОМАТИЗАЦІЇ ХМАРНОГО СЕРВІСУ

Текст програми

ЛИСТЗАТВЕРДЖЕННЯ

ГЮИК.502528.005 34 01- 01 12 01 – ЛЗ

РОЗРОБИВ

ст. гр. СПРм-19-2

Муренченко П.Є.

ЗАТВЕРДЖЕНО

ГЮИК.502528.005 34 01- 01 12 01 – ЛЗ

## СИСТЕМНИЙ ПІДХІД ДО АВТОМАТИЗАЦІЇ ХМАРНОГО СЕРВІСУ

Текст програмы

ГЮИК.502528.005 34 01 – 01 12 01

Листів 13

```
url.py
from django.contrib import admin
from django.urls import path, include
from users import views as userV

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('blog.urls')),
    path('reg/', userV.registration, name='reg'),
    path('auth/', userV.authorization, name='auth'),
    path('prof/', userV.profile, name='prof'),
    path('logout/', userV.log_out, name='lout'),
    path('market/', userV.market, name='market'),
    path('market/game/', userV.games, name='games')
    #path('admin/', admin.site.urls),
]

blog/views.py
from django.shortcuts import render
from django.http import HttpResponse
from django.contrib.sessions.models import Session
# Create your views here.

def session_check(request):
    if request.session.session_key:
        usern=Session.objects.get(session_key=request.session.session_key)
        user={ 'id': usern.get_decoded().get('id_user'), 'username':
usern.get_decoded().get('username'), 'role':
usern.get_decoded().get('role')}# дописать!!!
        return user
    else:
        return 0

def home(request):
    s_user=session_check(request)
    return render(request, 'blog/home.html', { 's_user': s_user })

def download(request):
    s_user=session_check(request)
    return render(request, 'blog/download.html', { 's_user': s_user })

def faq(request):
    s_user=session_check(request)
    return render(request, 'blog/faq.html', { 's_user': s_user })

def contacts(request):
    s_user=session_check(request)
    return render(request, 'blog/contacts.html', { 's_user': s_user })

blog/urls.py
```

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='blog-main'),
    path('download/', views.download, name='blog-download'),
    path('faq/', views.faq, name='blog-faq'),
    path('contacts/', views.contacts, name='blog-contacts'),
]

user/models.py
from django.db import models

# Create your models here.
class User(models.Model):
    id_user = models.AutoField(primary_key=True)
    login = models.CharField(max_length=45)
    password = models.CharField(max_length=45)
    email = models.CharField(max_length=45)
    birth_date = models.DateField()
    id_role = models.ForeignKey('UserRole', models.DO_NOTHING,
db_column='id_role', blank=True, null=True)
    summary = models.TextField(blank=True, null=True)
    discount = models.FloatField(blank=True, null=True)
    def __str__(self):
        return {'id_user':id_user, 'login':self.login, 'password':password,
'email':email, 'birth_date':birth_date, 'id_role':id_role, 'summary':summary,
'discount':discount}
    class Meta:
        managed = False
        db_table = 'user'

class UserRole(models.Model):
    id_user_role = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=45)

    class Meta:
        managed = False
        db_table = 'user_role'

class Developer(models.Model):
    id_developer = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=45)
    description = models.TextField(blank=True, null=True)
    foundation_date = models.DateField()
    dev_link = models.CharField(max_length=100, blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'developer'
```

```
class Game(models.Model):
    idgame = models.AutoField(primary_key=True)
    name = models.CharField(max_length=45)
    description = models.TextField()
    price = models.FloatField()
    date_release = models.DateField()
    id_developer = models.ForeignKey(Developer, models.DO_NOTHING,
db_column='id_developer')

    class Meta:
        managed = False
        db_table = 'game'

class GameHasGenre(models.Model):
    id_genre = models.ForeignKey('Genre', models.DO_NOTHING,
db_column='id_genre', primary_key=True)
    id_game = models.ForeignKey(Game, models.DO_NOTHING,
db_column='id_game')

    class Meta:
        managed = False
        db_table = 'game_has_genre'
        unique_together = (('id_genre', 'id_game'),)

class Genre(models.Model):
    idgenre = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=45)

    class Meta:
        managed = False
        db_table = 'genre'

class Order(models.Model):
    idorder = models.AutoField(primary_key=True)
    id_server = models.ForeignKey('Server', models.DO_NOTHING,
db_column='id_server')
    id_user = models.ForeignKey('User', models.DO_NOTHING,
db_column='id_user')
    date = models.DateTimeField()
    duration = models.FloatField()
    cost = models.FloatField()
    comment = models.TextField(blank=True, null=True)
    hash = models.CharField(max_length=45, blank=True, null=True)
    id_order_state = models.ForeignKey('OrderStatus', models.DO_NOTHING,
db_column='id_order_state', blank=True, null=True)
    id_type_service = models.ForeignKey('TypeService', models.DO_NOTHING,
db_column='id_type_service')
```

```
class Meta:
    managed = False
    db_table = 'order'

class OrderStatus(models.Model):
    id_order_status = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=45)

    class Meta:
        managed = False
        db_table = 'order_status'

class Server(models.Model):
    idserver = models.AutoField(primary_key=True)
    name = models.CharField(max_length=45)
    id_set = models.ForeignKey('SetOfEquipment', models.DO_NOTHING,
db_column='id_set')
    price = models.FloatField()
    id_serv_state = models.ForeignKey('ServerState', models.DO_NOTHING,
db_column='id_serv_state', blank=True, null=True)
    open_date = models.DateField()
    description = models.TextField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'server'

class ServerHasGame(models.Model):
    id_server = models.ForeignKey(Server, models.DO_NOTHING,
db_column='id_server', primary_key=True)
    id_game = models.ForeignKey(Game, models.DO_NOTHING,
db_column='id_game')

    class Meta:
        managed = False
        db_table = 'server_has_game'
        unique_together = (('id_server', 'id_game'),)

class ServerState(models.Model):
    id_server_state = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=45)

    class Meta:
        managed = False
        db_table = 'server_state'
```

```
class SetOfEquipment(models.Model):
    idset_of_equipment = models.AutoField(primary_key=True)
    name = models.CharField(max_length=45)
    date_setup = models.DateField()
    cpu_name = models.CharField(max_length=45)
    gpu_name = models.CharField(max_length=45)
    power_cpu = models.FloatField()
    ram_amount = models.IntegerField()
    storage = models.CharField(max_length=10)

    class Meta:
        managed = False
        db_table = 'set_of_equipment'

class TypeService(models.Model):
    idtype_service = models.IntegerField(primary_key=True)
    cost = models.FloatField()
    description = models.TextField()
    duration = models.CharField(max_length=45, blank=True, null=True)
    name = models.CharField(max_length=45)

    class Meta:
        managed = False
        db_table = 'type_service'

user/views.py
from django.shortcuts import render, redirect
from django.http import HttpResponse
from django.contrib.sessions.models import Session
from django.contrib.sessions.backends.db import SessionStore
from django.contrib.auth.forms import UserCreationForm
from django.contrib import messages
from .models import User
from .models import UserRole
from .models import Game
from .models import Server
from .models import Developer
from .models import GameHasGenre
from .models import Genre
from .models import Order
from .models import OrderStatus
from .models import ServerHasGame
from .models import ServerState
from .models import SetOfEquipment
from .models import TypeService
from django.db import connection
from django.core.paginator import Paginator
import re

def session_check(request):
```

```
    if request.session.session_key:
        usern=Session.objects.get(session_key=request.session.session_key)
        user={ 'id': usern.get_decoded().get('id_user'), 'username':
usern.get_decoded().get('username'), 'role': usern.get_decoded().get('role')}
        return user
    else:
        return 0

# Create your views here.
def registration(request):
    if request.method=="POST":
        if re.match('^[A-Za-z0-9_]{4,44}$', request.POST['username']):
            username=request.POST['username']
            if User.objects.filter(login=username).exists():
                messages.warning(request, f'This username already
registered')
                return render(request, 'users/registration.html')
            else:
                messages.warning(request, f'Wrong username')
                return render(request, 'users/registration.html')

        if re.match('^[A-Za-z0-9_]{4,20}@[a-z]{3,15}\.[a-z]{2,5}$',
request.POST['e_mail']):
            email=request.POST['e_mail']
            if User.objects.filter(email=email).exists():
                messages.warning(request, f'This email already
registered')
                return render(request, 'users/registration.html')
            else:
                messages.warning(request, f'Wrong e-mail. Pattern:
example@gmail.com')
                return render(request, 'users/registration.html')

        if re.match('^[A-Za-z0-9_]{4,44}$', request.POST['pass']):
            passw=request.POST['pass']
        else:
            messages.warning(request, f'Wrong password')
            return render(request, 'users/registration.html')

        if passw==request.POST['conf_pass']:
            dateB=request.POST['dateB']
            b=User(login=username, password=passw, email=email,
birth_date=dateB, id_role=UserRole.objects.get(id_user_role=1))
            b.save()
            messages.success(request, f'Registration success!')
            return redirect('blog-main')
        else:
            messages.warning(request, f'Wrong confirm password')
            return render(request, 'users/registration.html')

def authorization(request):
```

```

    if request.method=="POST":
        if 'username' in request.POST:
            username=request.POST['username']
            passw=request.POST['pass']
            messages.success(request, f'{username} try login with
pass={passw}')
            if User.objects.filter(login=username).exists(): #Проверка на
существование
                f_user=User.objects.get(login=username)
                if f_user.password==request.POST['pass']:
                    messages.success(request, f'{username} login')
                    request.session['username']=username

            request.session['role']=f_user.id_role.id_user_role
            request.session['id_user']=f_user.id_user
            return redirect('blog-main')
            else:
                messages.warning(request, f'Wrong username or
password')
            else:
                messages.warning(request, f'Wrong username or
password')
            else:
                messages.warning(request, f'Empty login field')
            return render(request, 'users/authorization.html')

def profile(request):
    if request.session.session_key:

        usern=Session.objects.get(session_key=request.session.session_key)
        us=usern.get_decoded()
        return render(request, 'users/registration.html')

def log_out(request):
    if request.session.session_key:
        request.session.flush()
        return redirect('blog-main')

def market(request):
    game=Game.objects.all()
    paginator=Paginator(game, 10)
    if request.method=="GET":
        if 'page' in request.GET:
            page=int(request.GET['page'])
        else:
            page=1
        if 'service' in request.GET:
            pass
        if 'g_search' in request.GET:
            pass
    if request.method=="POST":

```

```

        pass
        s_user=session_check(request)
        #(idgame__gte=1, idgame__lte=10)#[0,10]
        last_page=paginator.num_pages
        prev_page=page-1
        next_page=page+1
        games=paginator.get_page(page)
        service=TypeService.objects.all()
        return render(request, 'users/market.html', { 's_user': s_user ,
's_curent_page': page, 'last_page': last_page, 'next_page': next_page,
'prev_page': prev_page, 'games':games, 'services': service})

def games(request):
    s_user=session_check(request)
    if request.method=="GET":
        if 'more' in request.GET:
            id=request.GET['more']
            servers=[]
            game=Game.objects.get(pk=id)
            for ids in ServerHasGame.objects.filter(id_game=id):
                servers.append(ids.id_server)
            return render(request, 'users/games.html', { 's_user':
s_user, 'games':game, 'servers': servers,'action': 1})
        if request.method=="POST":
            if 'editView' in request.POST:
                servers=[]
                idgame=request.POST['editView']
                game=Game.objects.get(pk=idgame)
                for ids in ServerHasGame.objects.filter(id_game=id):
                    servers.append(ids.id_server)
                return render(request, 'users/games.html', { 's_user':
s_user, 'games':game, 'servers': server, 'action': 2})
            if 'editAccept' in request.POST:
                game=Game.objects.get(pk=request.POST['editAccept'])
                game.name = request.POST['name']
                game.description = request.POST['description']
                game.price = request.POST['price']
                game.date_release = request.POST['date_release']
                game.id_developer =
Developer.objects.get(pk=request.POST['id_developer'])
                game.save()
                messages.success(request, f'Edited successfully')
                return redirect('market')
            if 'delete' in request.POST:
                game=Game.objects.get(pk=request.POST['delete'])
                return render(request, 'users/games.html', { 's_user':
s_user, 'games':game, 'action': 3})
            if 'accept' in request.POST:
                game=Game.objects.get(pk=request.POST['accept'])
                game.delete()
                messages.success(request, f'Deleted successfully')

```

```

        return redirect('market')
    if 'cancel' in request.POST:
        return redirect('market')

def search(request):
    s_user=session_check(request)
    if request.method=="GET":
        if request.GET('name'):

            games=Game.objects.filter(name__contains=request.GET('name'))
            return render(request, 'users/market.html', { 's_user': s_user ,
'games':game})

def order(request):
    s_user=session_check(request)
    return render(request, 'users/orders.html', { 's_user': s_user ,
'games':game})

market.html
{% extends 'blog/main.html' %}
{% load staticfiles %}
{% block main %}
<div class="plateContainer">
    <h2>Услуги</h2>
    {% for service in services %}
    <div class="plate">
        <h3>{{service.name}}</h3>
        <p style="height: 63%;">{{service.description}}</p>
        <form action="{% url 'market' %}" method="GET">
            <button type="submit" style="margin-left:100px"
name="service" value="{{service.idtype_service}}">Заказать</button>
        </form>
    </div>
    {% endfor %}
</div>
<div>
    <h2>Игры</h2>
    <form style="margin-bottom:20px">
    <p><input type="search" name="q" placeholder="Поиск игр">
    <button type="submit" name="search">Найти</button>
    {% if s_user and s_user.role == 2 %}<button type="submit"
name="addgame">Добавить игру</button>{% endif %}
    </p>
    </form>

    {% for game in games %}
        <div class="game_container">
            <div class="img_container">
                
            </div>

```

```

        <div class="text_container">
            <b><p>{{ game.name }}</p></b>
            <p>{{ game.description }}</p>
        </div>
        <div class="button_container">
            <form action="{% url 'games' %}" method="GET">
                <button type="submit" {% if s_user and s_user.role
==          2          %}>style="margin:1px" {%
value="{{game.idgame}}">Подробнее</button>
                </form>
                {% if s_user and s_user.role == 2 %}
                <form action="{% url 'games' %}" method="POST">
                    {% csrf_token %}
                    <button type="submit" style="margin:1px;
background-color:orange"
value="{{game.idgame}}">Изменить</button>
                    <button type="submit" style="margin:1px;
background-color:red" name="delete" value="{{game.idgame}}">Удалить</button>
                </form>
                {% endif %}
            </div>
        </div>
    {% endfor %}
    <form action="{% url 'market' %}" method="GET">
        {% if curent_page > 1 %}<button type="submit" name="page"
value="1">&laquo;</button>
        <button type="submit" name="page"
value="{{prev_page}}">&lt;</button>{% endif %}
        {{curent_page}}
        {% if curent_page < last_page %}<button type="submit" name="page"
value="{{next_page}}">&gt;</button>
        <button type="submit" name="page"
value="{{last_page}}">&raquo;</button>{% endif %}
    </form>
</div>
{% endblock main %}

registration.html
{% extends 'blog/main.html' %}
{% block main %}
<form class="logform" action="{% url 'reg' %}" method="POST">
    {% csrf_token %}
    <p>Username</p>
    <input class="my-1" type="text" name="username"><br>
    <p>E-mail</p>
    <input class="my-1" type="text" name="e_mail"><br>
    <p>Password</p>
    <input class="my-1" type="password" name="pass"><br>
    <p>Confirm password</p>
    <input class="my-1" type="password" name="conf_pass"><br>
    <p>Date birth</p>

```

```

    <input class="my-1" type="date" id="start" name="dateB" value="2000-
01-01" min="1930-01-01" max="2006-12-31"><br>
    <button type="submit">Registration</button>
</form>
<p>{{ users }}</p>
{% endblock main %}

authorization.html
{% extends 'blog/main.html' %}
{% block main %}
<form class="logform" action="{% url 'auth' %}" method="POST">
    {% csrf_token %}
    <p>Username</p>
    <input class="my-1" type="text" name="username"><br>
    <p>Password</p>
    <input class="my-1" type="password" name="pass"><br>
    <button type="submit">Log in</button>
</form>
{% endblock main %}

games.html
{% extends 'blog/main.html' %}
{% load staticfiles %}
{% block main %}
{% if action == 1 %}
<h2>{{games.name}}</h2>
<p>Жанр:{{games}}</p>
<p>Разработчик: <a target="_blank" rel="noopener noreferrer"
href="{{games.id_developer.dev_link}}">{{games.id_developer.name}}</a></p>
<p>Дата релиза: {{games.date_release}}</p>
<p>Описание:</p>
<p>{{games.description}}</p>
<h2>Сервера с установленной игрой:</h2>
{% for server in servers %}
<p>{{server.name}}</p>
<p>Процессор:{{server.id_set.power_cpu}}GHz</p>
{% endfor %}
{% endif %}
{% if action == 3 %}
<h2>{{game.name}}</h2>
<p>Данная информация об игре будет удалена. Вы уверены?</p>
<form action="{% url 'market' %}" method="POST">
    {% csrf_token %}
    <button type="submit" style="margin:1px; background-color:red"
name="accept" value="{{game.idgame}}">Подтвердить</button>
    <button type="submit" style="margin:1px; background-color:green"
name="cancel" value="{{game.idgame}}">Отклонить</button>
</form>
{% endif %}

{% endblock main %}

```

