**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**
**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

**Кафедра математичних методів захисту інформації**

# ЗВІТ

## З ВИРОБНИЧОЇ ПРАКТИКИ

Напрям підготовки: 6.040301 «Прикладна математика»

Тема: «Безпека розподілених геоінформаційних систем»

Виконав студент 3 курсу

групи ФІ-13

Кригін Валерій Михайлович

Керівник:

д.т.н. Куссуль Наталія Миколаївна

_____

*(підпис)*

Захистив з оцінкою:

_____

**Київ 2014**

# ЗМІСТ

# ПЕРЕЛІК ТЕРМІНІВ ТА УМОВНИХ ПОЗНАЧЕНЬ

**BPMN** (англ. Business Process Model and Notation, нотація та модель бізнес-процесів) система умовних позначень для моделювання бізнес-процесів.

**Абстракція** узагальнення більш простих понять до більш складних, розглядання конкретного явища замість видів, в яких воно може поставати.

**База знань** особливого роду база даних, розроблена для управління знаннями (метаданими), тобто збором, зберіганням, пошуком і видачею знань. Використовується в експертних системах.

**Бібліотека (програмування)** збірка об'єктів чи підпрограм для вирішення близьких за тематикою задач.

**Експертна система** Комп'ютерна система, що здатна частково замінити експерта.

**Класифікація** система розподілення об'єктів (процесів, явищ) за класами (групами тощо) відповідно до визначених ознак.

**Кластерний аналіз (кластеризація)** задача розбиття заданої вибірки ситуацій (об'єктів) на підмножини, що називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Частковий випадок класифікації.

**Комп'ютерне навчання** Навчання людей за допомогою комп'ютера.

**Машина виведення** програма, яка виконує логічний вивід з попередньо побудованої бази фактів і правил згідно з законами формальної логіки.

**Машинне навчання** підрозділ штучного інтелекту, що вивчає методи побудови моделей, що здатні до самонавчання.

**Прецедент** специфікація послідовності дій при проектуванні програмних систем.

**Програмний фреймворк** готовий до використання комплекс програмних рі-

шень, включаючи, дизайн, логіку та базову функціональність системи або підсистеми.

# ВСТУП

*Об'єкт дослідження*: користувачі, експертні системи, їх взаємодія.

*Предмет дослідження*: поведінка користувачів при навчанні, реакція на різноманітний зовнішній вплив.

Окремий інтерес собою являє правильність трактування даних, що отримуються від користувачів в результаті взаємодії з експертною системою, — важливіша і складніша частина роботи, що і відрізняє її від аналогів.

системи для найбільш ефективного процесу навчання.

*Метою роботи є* побудова експертної системи комп'ютерного навчання користувачів на прикладі обробки геоінформаційних даних. Така задача є достатньо багатогранною, для її втілення потрібно працювати з даними різного походження, також природньо виникає потреба використання розподіленої системи. На такій базі буде розроблено достатньо загальний метод навчання, що використовуватиметься для інших дисциплін.

Основна задача — мінімізувати витрати часу експерта (викладача) на процес перевірки знань користувачів (студентів) та надання їм навчальних матеріалів, підвищення об'єктивності оцінювання користувачів, а також оцінка роботи самих викладачів. Це буде зроблено шляхом розробки експертної системи, що буде аналізувати дії користувачів, класифікувати їх, робити відповідні висновки.

Тут постає питання інформаційної безпеки, адже потрібно контролювати користувачів — перевіряти, чи та людина сидить за своїм робочим місцем, запобігати списування, гарантувати максимальну безпеку базі даних з результатами робіт, а також стежити за тим, щоб користувачі отримували коректні завдання, а розв'язки відправлялися системі в незміненому вигляді.

# 1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

## 1.1 Власне, завдання

Як вже було описано вище, індивідуальне завдання — створення розподіленої геоінформаційної системи, призначення якої — навчання обробці і використанню даних, що фігурують в таких системах. Це є векторні та растрові дані, таблиці, тексти, числа, діаграми і таке інше — тобто, дані різноманітного походження, які потребують застосування різних методів зберігання та обробки, але треба створити єдину систему, що буде коректно працювати з усім цим розмаїттям єдиним способом.

Далі ця система буде розширена і узагальнена на інші галузі і інші потреби — не тільки навчання студентів, але й навчання та підвищення кваліфікації персоналу підприємств, і не тільки для використання та обробки геоданих.

## 1.2 Актуальність проблеми

В світі інформаційних технологій жити стає дедалі зручніше. Завдяки потужним комп’ютерам і мережі Інтернет стає можливою передача великих обсягів інформації за відносно малий час. Чому б не використати можливості сучасних інформаційних технологій для покращення процесу навчання?

Різні викладачі мають різну методику викладання, різні вимоги до студентів, по-різному запобігають списуванню. Як щодо того, щоб об’єднувати знання і навички викладачів, обирати найефективніші?

Підемо далі! Можна створити самодостатню систему, що зможе самостійно наглядати за студентами та синтезувати нові правила класифікації користувачів, щоб викладач міг приділяти більше часу важливим консультаціям, коректівам

до курсу, створенню та внесенню змін до практичних завдань, або ж просто на відпочинок.

## 1.3 Аналоги

Перші кроки вже зроблено — існують електронні щоденники шкільників, електронні конспекти з різних дисциплін, електронні тести для перевірки знань. На мою думку, найбільш розвиненою з таких систем є Coursera [1].

Coursera надає доступ до навчальних матеріалів (відеолекцій, друкованих конспектів), проводить перевірку знань у вигляді тестів. Зокрема проект має дуже цікаву можливість підтвердження особистості за допомогою веб-камери та електронного почерку, що дозволяє видавати сертифікати за проходження курсів, яким можна довіряти. Взаємне оцінювання студентів у спірних питаннях також є продуктивною ідеєю, що дозволить підвищити об’єктивність оцінки рівня складності завдань та внесе різноманіття в процес навчання.

Чого ж нам не вистачає?

### 1.3.1 Суб’єктивне уявлення системи про користувача

Звісно, студентів ніщо не переможе, але можна почати робити перші кроки в аналізі поведінки користувачів математичними методами.

Можна будувати емпіричні моделі — сказати, що цей и цей студент поганий, тому їх поведінка — поведінка студентів, що списують. Або інакше — створити правило, згідно з яким студент, що погано виконував практичні завдання, а на екзамені показав гарні результати, обов’язково списав. Це не є те, що нам потрібно, але від таких спостережень можна відштовхуватись.

Тобто, кінцевий продукт повинен аналізувати дії користувачів, результати

тестування та інші фактори, щоб окрім оцінки правильності виконання робіт давати ще оцінку чесності отриманих результатів.

### 1.3.2 Вибір завдань для кожного користувача в індивідуальному порядку

Люди спілкуються один з одним і намагаються одне одному допомогти (або заважати) — це природньо.

Розглянемо типічну ситуацію: хтось проходить тест, фотографує завдання, і вже на наступний раз (на випадок необхідності переписати роботу) у кількох студентів вже є приклад завдання (а може навіть розв'язаний).

Навіть якщо викладач має багато варіантів завдань, не виключається можливість того, що один і той самий студент кілька разів отримає один і той самий білет.

Що робити викладачу? Записувати, який студент які завдання виконував? Дізнаватися, хто з ким товарищує і хто з ким напевно ділиться розв'язками? На мою думку, це дуже складно і не варто таких зусиль.

Тим не менш, для комп'ютерної системи такі обмеження майже знімаються. Наприклад, база даних, в якій містяться дані про те, хто, коли і як виконував певні завдання, а на основі цієї інформації приймати рішення щодо тесту, який дати певному користувачу. Комп'ютерам можно призначити ідентифікатори, щоб знати, хто сидить по сусідству з даним користувачем (з приватними комп'ютерами все складніше, але можна щось запропонувати). Інтеграція системи з соціальними мережами дасть додаткову інформацію про зв'язки у групах.

Також можуть бути користувачі, що претендують на невисокий бал — для них можна намагатися шукати прості завдання, виконання яких не дасть високої оцінки, але яке вважається достатнім для того, щоб студент продемонстрував

свої знання. Навпаки, для студентів, що подають великі надії, можна обирати задачі підвищенної складності, що можуть дати додаткові бали.

### 1.4 Прецеденти

Послідовність дій студента за курс навчання якогось предмету можна розкласти на наступні етапи:

1) Записатися на курс (вступити до університету)
2) Ознайомитися з теорією (слухати лекції)
3) Виконати практичні завдання для закріплення теорії (виконання домашніх завдань, контрольних робіт)
4) Повторювати 2 і 3 за навчальним планом
5) Продемонструвати свої знання та навички у зазначений строк (залік, екзамен, курсова)

Послідовність дій викладача при навчанні:

1) Стати викладачем
2) Створити план курсу — план лекцій, практичних занять тощо
3) Ознайомити студентів з курсом (провести лекції)
4) Робити проміжні перевірки знань (контрольні, домашні завдання)
5) Зробити кінцевий контроль знань (екзамен, залік)

З першого погляду схема здається простою, але обов’язково потрібно враховувати людський фактор — недобросовісними можуть бути як студенти, так і викладачі. Студентів багато, а викладач один, тому списування стає справжньою проблемою, яку різні викладачі вирішують по-різному, але основа одна — вияв-

лення основних ознак списування та впровадження штрафних санкцій.

Розглянемо більш детально процес проміжної перевірки знань (написання контрольних робіт), точніше, огляд схеми, яка запропонована авторами роботи як еталонна:

1) Викладач займає своє робоче місце, впускає студентів
2) Кожен студент займає своє робоче місце
3) Кожен студент отримує індивідуальне завдання
4) Викладач (асистент) записує, який студент які завдання отримав
5) Студенти приступають до розв'язання своїх задач
6) Викладач (асистент) слідкує за тим, щоб студенти не списували
7) У разі списування викладач ставить примітку з вказанням ступені, орієнтованих обсягів списування в журналі, та ліквідує подальшу можливість даного студента скористатися джерелом списування — забирає шпаргалку, відсаджує від сусіда тощо
8) Коли студент вирішив, що впорався з завданням як міг, він здає роботу і більше не має можливості її редагувати
9) Коли витікає строк складання тесту (контрольної), викладач (асистент) збирає те, що виконали студенти, або просить здати роботи самостійно
10) Роботи роздаються студентам на перевірку (свою роботу студент перевіряти не може), якщо це передбачено для даної роботи
11) Студенти перевіряють роботи так, щоб можна було відрізнити те, що написано автором роботи, від того, що додав перевіряючий. Якщо це не виявляється неможливим (наприклад, студент використав усі можливі кольори ручок, олівців, тому відрізнити його роботу від приміток перевіряючого неможливо), робота одразу потрапляє на перевірку до викладача
12) Асистент стежить за чесністю перевірки робіт студентами
13) Коли закінчується строк перевірки, роботи здаються

14) По закінченню заняття студенти залишають приміщення
15) Після студентів приміщення залишає і сам викладач (з асистентом)
16) У вказані строки викладач перевіряє роботи студентів, виставляє оцінки. Також викладач може залишати суб'єктивні примітки щодо чесності написання роботи і в залежності від них змінювати бали
17) У зазначений час (наприклад, під час практичних занять) студенти повинні ознайомитися з результатами
18) Якщо хтось не згоден з результатами, він може попросити проглянути свою роботу
19) Викладач (асистент) повинен стежити, щоб студенти не виправляли роботу під час проглядання
20) Якщо студент довів (або не довів) свою правоту, оцінка може змінитися. Також йому може бути дана робота на переписування (з іншими завданнями)
21) Для кожної роботи має бути лімітована кількість разів переписування

Тобто, на кожному етапі треба слідкувати за тим, щоб студенти мали справу лише зі своєю роботою, не мали змоги виправити написану роботу, не могли змінити результати роботи в журналі викладача, але все це досить тривіальні, на мою думку, речі, якими з давніх давен займається захист інформації.

Інновації відбуваються на більш абстрактному рівні — на рівні знань студентів і співставлення їх з виконаними роботами, що має викладач. Якщо все ж таки мало місце списування, потрібно це виявити, додати характерні ознаки "місця злочину" до бази знань та на її основі робити подальші висновки щодо інших студентів. Є ще цікавий випадок — коли студенти, що мають знання, перехвилювалися і наробили помилок. Звісно, все це аналізувати складно, але спробувати варто — не можна стояти на місці.

# 2 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Одним з найбільш повних джерел за тематикою автоматизації комп'ютерного навчання є книга Булигіна В.Г. "Основы автоматизации процесса обучения" [2]. В ній досить повно розглядається проблема автоматизації навчання — від математичних моделей окремих компонент до фізичної реалізації цілої загальнодержавної системи, мета якої — покращення якості навчання взагалі. Тим не менш, система, якій присвячено даний звіт, є складною, і необхідні для її створення знання виходять за межі книги, хоч і переслідує все ту ж мету — покращення і автоматизація процессу навчання. Наприклад, задачі створення рекомендацій для викладача, аналізу практичних робіт студентів на предмет нечесної здачі тощо, є важливими, але так і не були висвітлені в книзі.

Після поверхневого огляду задач системи та методів, якими вони повинні розв'язуватись, було виділено наступні галузі знань, які потрібно більш детально розібрати для досягнення поставленої мети:

1) **Математичні методи прийняття рішень** для багатокритеріальних альтернатив. Дана галузь охоплює задачу вибору відповідних завдань для конкретного студента з урахуванням багатьох критеріїв, кількість яких може змінюватись як в рамках конкретного завдання, так і в залежності від даних, які є для конкретного студента. Наприклад, якщо студент вже виконував задачу $n$, то система повинна знайти для нього інше завдання, аби підвисити об'єктивніть оцінювання.

   Основним джерелом за темою прийняття рішень на момент написання звіту є курс Смірнова С.А. "Моделі та методи прийняття рішень", що дав базове бачення предметної області.
2) **Методи машинного навчання** в реалізації системи необхідні для аналізу результатів завдань і створення необхідних рекомендацій щодо покращен-

ня рівня навчання. Планується аналізувати як користувачів (успішність та ступінь чесності студентів, об'єктивність оцінювання перевіряючих осіб), так і складність самих тестових завдань, щоб зробити оцінювання якомога більш об'єктивним.

Для ознайомлення з деякими методами були використані матеріали ІКД для внутришнього користування та інформація з сайту MachineLearning.Ru [3]. Дана тема потребує більш детального вивчення перед тим, як будуть обрані конкретні методи та алгоритми машинного навчання.

Під час проходження практики особливу увагу було приділено алгоритмам кластерізації k-means та k-means++, в задачі обробки супутникових знімків. Були використані як вже готові готові реалізації даних алгоритмів в середовищі MatLab, так і окремі реалізації, розроблені спеціально для задачі обробки супутникових знімків.

3) **Безпека розподілених систем** займає важливе місце в даній роботі, адже сама система є розподіленою і передбачає використання декількох серверів для залучення різноманітних тестових даних. Також серед студентів можуть бути особи, що можуть спробувати нечесно змінити свої результати, знищити дані або систему, що змусить витратити додатковий час на відновлення системи, заново проводити тестування, а в крайньому разі унеможливить подальше використання системи за призначенням.
4) **Математична статистика** грає важливу роль у створенні моделей студентів та викладачів. Система отримуватиме результати роботи користувачів, аналізуватиме їх в купі та буде розподіляти їх за певними крітеріями. Коли статистичні методи не будуть допомагати, потрібно буде прибігати до нейронних мереж для вирішення задачі класифікації.

# 3 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО МЕТОД РОЗВ'ЯЗАННЯ, ЙОГО ОБҐРУНТУВАННЯ

Основа реалізації проекту — розподілена експертна система. Зрозуміло, що для цього потрібно використовувати мови програмування, які дозволяють зручно працювати з розподіленими системами, а також мають якомога кращу підтримку експертних систем — готові фреймворки, бібліотеки.

Увагу зачепили дві мови — Erlang і Scala, коротке представлення про які наведено далі.

## 3.1 Erlang

Erlang — перевірена часом мова програмування, що була створена у 1986 році, і розвивається до наших днів. Створений компанією Ericsson спеціально для розподілених систем з великим навантаженням. Якщо в інших мовах для підтримки багатопотоковості потрібно завантаження спеціальних бібліотек, Erlang має ці можливості як основні — на рівні мови.

Основа розподіленої роботи в Erlang — обмін повідомленнями між програмами, що дозволяє розподіляти обробку даних як між різними процесами в межах одного комп'ютеру, так між різними серверами, полегшується обробка виключних ситуацій тощо.

Одним з фреймворків для створення експертних систем є ERESYE — ERlang Expert SYstem Engine (движок для створення експертних систем на Erlang) [4].

ERESYE — це бібліотека для створення, виконання та управління машинами виведення [5], що й треба для створення експертних систем.

## 3.2 Scala

Одним з варіантів платформи реалізації описуваної системи є мова програмування Scala і відповідно віртуальна машина JVM. Переваги використання цього варіанту:

1) Великий вибір готових бібліотек та можливість виконуватись будь-де (дану можливість реалізує платформа JVM).
2) Широкий вибір програмного інструментарію для розробки розподілених програм.
3) Наявність багатьох готових фреймворків для реалізації експертних систем, серед яких:

    а) Drools [6] — перевірена часом бібліотека, що має відкритий програмний код та велику спільноту розробників, що дозволяє швидко знайти відповідь на будь-яке питання її стосовно функціональності.
    б) D3web [7] — більш проста бібліотека зі своїм Wiki-ресурсом та наявністю юніт-тестування. Більш просте рішення, яке, можливо, краще підходить для рішення задачі, описаної в даному звіті.
    в) jColibri [8] — бібліотека, яка призначена для інтерактивного пошуку по великому числу варіантів.

З вищенаписаного можна зробити висновок, що Scala чудово підходить для реалізації ідей, описаних в наступному підрозділі.

# 4 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНИХ АЛГОРИТМІВ

Оскільки часу було замало, а ідея нова, програмної реалізації ще немає. Замість цього в даному розділі буде наведено основні ідеї та діаграми.

У підрозділі 1.4 можна побачити послідовності дій студентів та викладачів у реальному житті. Задача даної частини доповіді — представлення архітектури майбутньої системи.

Розглянему діаграму на рис. 4.1, на якій представлено загальну структуру системи — її компоненти та зв'язки між ними.

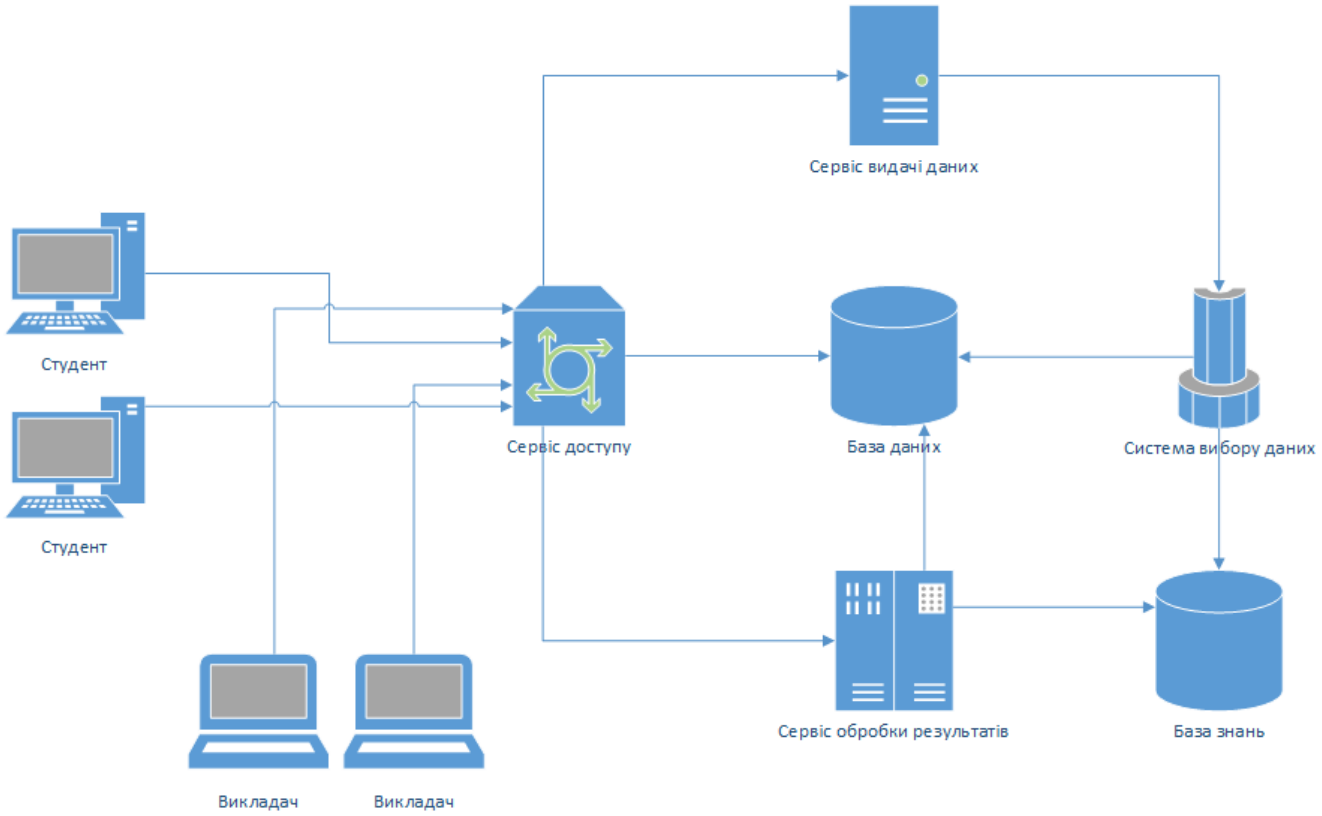


Рисунок 4.1 — Схема роботи системи

Тепер подивимось на схему на рис. 4.2.

Користувачам потрібно спочатку авторизуватися, після чого студенти зможуть отримувати завдання для виконання від відповідного сервера (сервіс видачі даних для тестування), а викладачі зможуть отримувати від системи результати

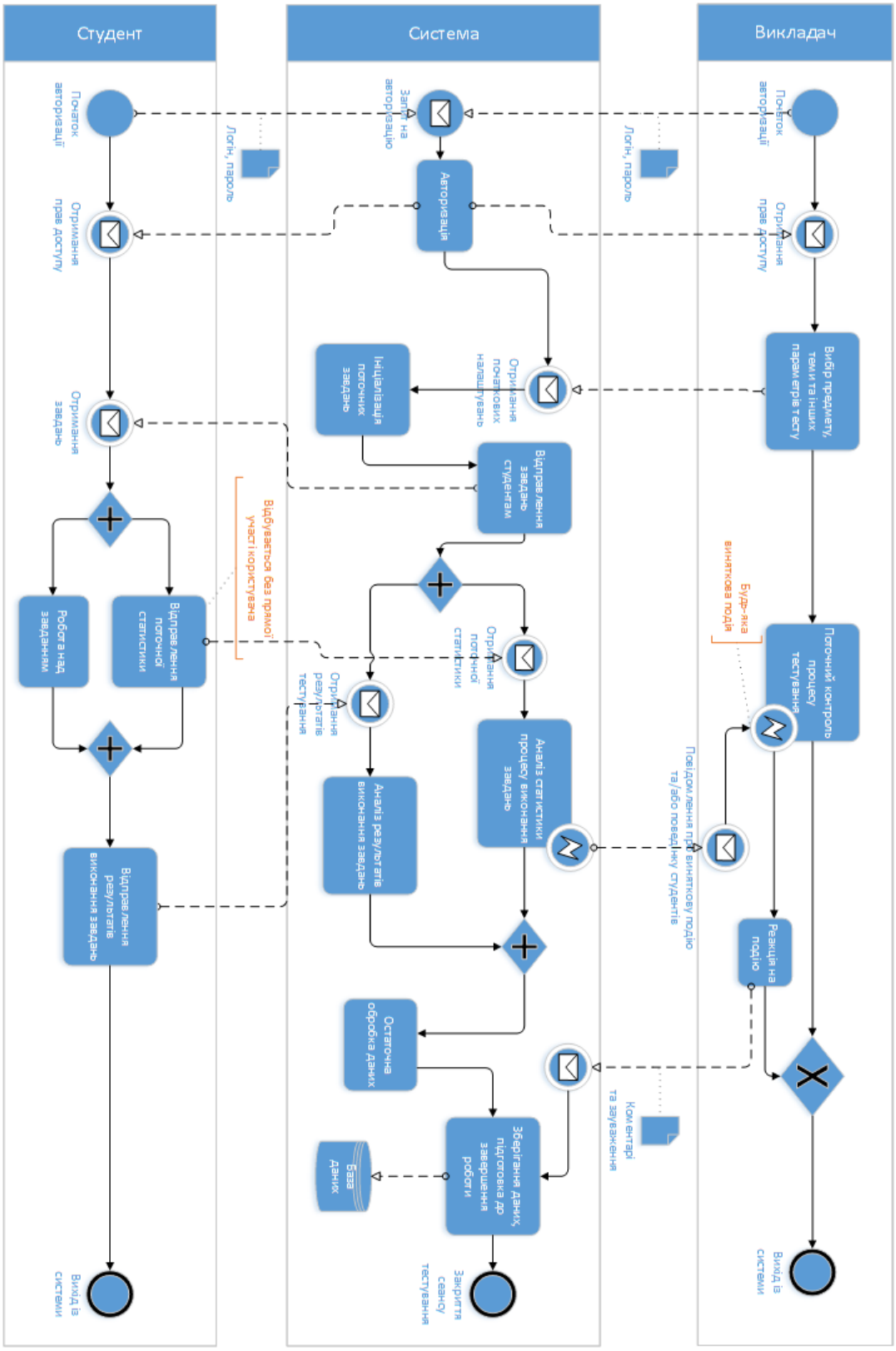


Рисунок 4.2 — BPMN-діаграма процесу проміжної перевірки знань, побудована на базі підрозділу 1.4

і ряд рекомендацій щодо оцінювання.

Сервіс видачі даних орієнтується на ряд правил, серед яких частота користування даними, використані дані в конкретній групі студентів, рівень знань конкретного студента, рівень складності завдань тощо. Правила націлені на покращення об’єктивності оцінювання.

Сервіс обробки результатів тестування опрацьовує результати кожного оцінювання. При цьому відбувається як перевірка правильності виконання роботи, так і збір різного роду статистики (час виконання як всього завдання так і окремих підзавдань, кількість виправлень тощо), на основі якої система може порекомендувати викладачу подальші дії: провести повторне тестування конкретного студента, провести усне опитування (якщо студент не викликає довіри протягом декількох спроб), змінити складність завдань або взагалі переробити/видалити завдання.

Також можна розбити викладачів на декілька рівнів ієрархії відповідальності, де інформація про оцінювання викладачами нижчих рівнів буде передаватися викладачам вищих рівнів ієрархії.

Система не орієнтується на конкретний тип даних, тому можуть використовуватись як прості тести з набором готових відповідей, так і завдання, де необхідно написати текст пояснення, код програми, намалювати контур, побудувати діаграму тощо.

Розглянемо схему з рис. 4.3. Вона описує поведінку системи в режимі функціонування система-викладач. Опустивши деталі авторизування користувачів, будемо вважати, що певну кількість тестів успішно завершено (є тестову вибірку) і система має дані для роботи.

За таких початкових умов викладачі, що мають доступ до даних конкретного курсу і групи, можуть продивлятися успішність студентів та аналізувати статистику процесу тестування, що представлена наглядно. На основі доступних даних викладач може змінювати оцінку студенту та приймати рішення щодо

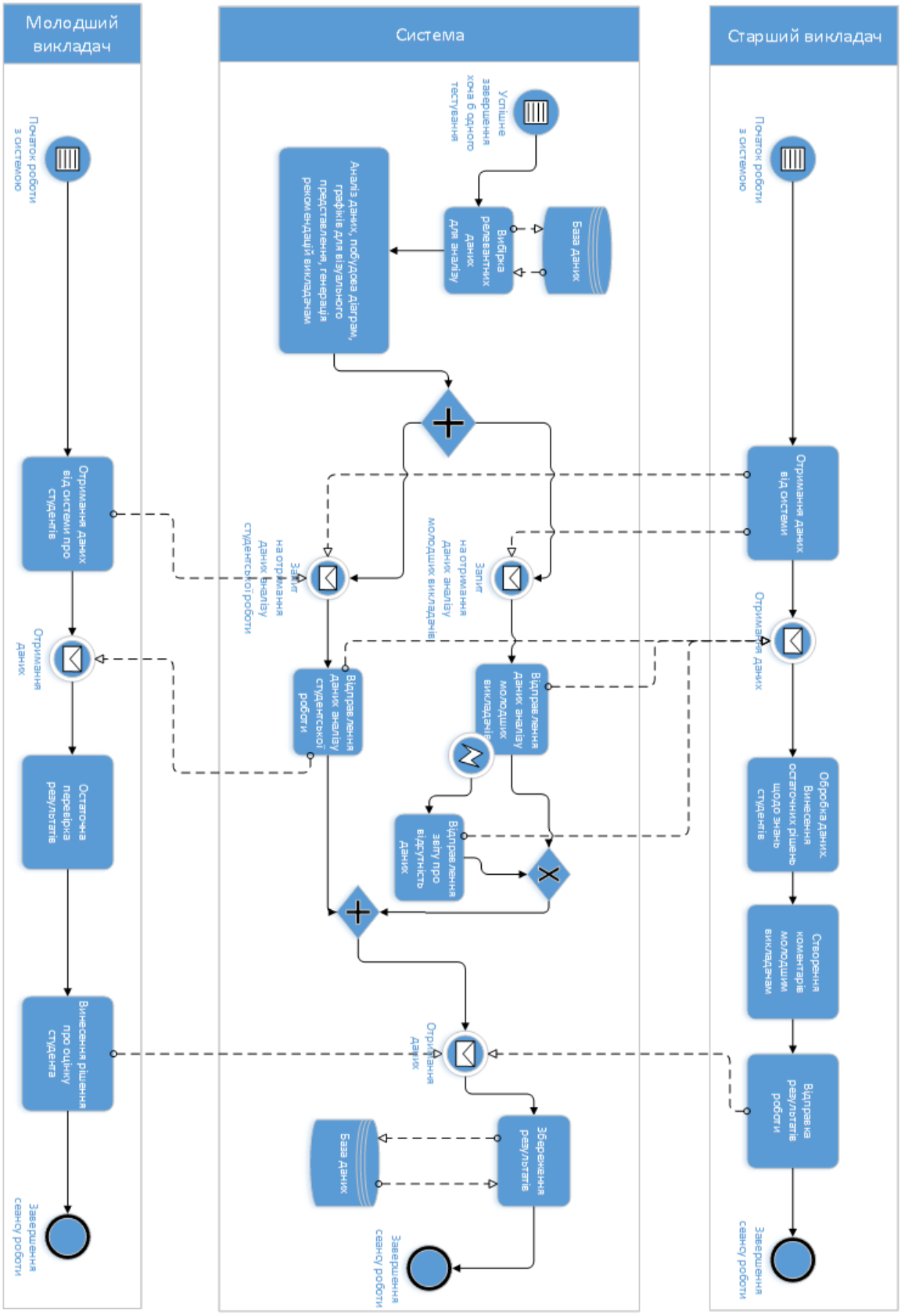


Рисунок 4.3 — BPMN-діаграма процесу взаємодії викладачів і системи

подальшого оцінювання.

При реалізації системи онлайн тестування, тобто, в таких умовах, де студент може здавати завдання з будь-якого місця, є доцільним реалізувати можливість викладачу надати новий блок завдань студенту та відправити йому повідомлення зі своїм рішенням — в тому числі про необхідність усної здачі.

Крім того, викладач вищого рівня може контролювати оцінювання викладачів нижчого. Наприклад, наскільки швидко завдання остаточно перевірені, наскільки об’єктивне оцінювання та багато іншого.

Як було сказано у вступі, дана частина системи є однієї із найважливіших, бо саме вона є базою для ідеї покращення об’єктивності навчання шляхом глибокого аналізу різноманітних даних.

# ВИСНОВКИ

Під час практики було розроблено базову концепцію майбутньої системи, проаналізовано кілька готових рішень та фреймворків для розробки експертних систем, були створені базові вимоги до системи, проведено аналіз проблеми автоматизації навчання та створено ряд ідей щодо його покращення. З використанням даних ідей буде створено реальну систему покращення навчання з мінімальними затратами часу та коштів.

Щодо вибору мови програмування, на якій буде реалізовано проект: JVM має великий набір бібліотек і більш широке коло розробників, але Erlang націлений на створення розподілених систем з високими вимогами до безпеки, а перевірка часом і достатня кількість готових систем, що працюють роками, дозволяють йому бути серйозним конкурентом інших платформах у сфері розподілених обчислень.

Також було вирішено, які галузі математики потрібні для досягнення поставленої мети — побудови моделей для максимально об’єктивного оцінювання, що означає захист достовірності отримуваної від студентів інформації.

Маємо плани на світле майбутнє, засоби та знання, ідеї щодо реалізації — залишилося з користю витратити наданий час.

Ідея є дійсно актуальною та варта реалізації.

# ПЕРЕЛІК ПОСИЛАНЬ


1. *Stanford University*. Coursera. — https://www.coursera.org. — 2014.

2. *Булыгин, В. Г.* Основы автоматизации процесса обучения / В. Г. Булыгин. — Йошкар-Ола, 2003. — 190 с.

3. Machinelearning.ru. — http://www.machinelearning.ru. — 2014.

4. *Typed Lambda*. Eresye. — https://github.com/TypedLambda/eresye. — 2014.

5. *Arsanukaev, Yasir M.* Искусственная разумность в erlang: транспортная область. — http://kingping.blinkenshell.org/Expert_system_in_Erlang__the_Domain_of_Transport.html.

6. *JBoss Community*. Drools. — http://drools.jboss.org. — 2014.

7. d3web. — http://d3web.de. — 2014.

8. *GAIA*. jcolibri. — http://gaia.fdi.ucm.es/research/colibri/jcolibri. — 2014.