

Н.Б. Шаховська, Д.М. Козловський
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ІНТЕЛЕКТУАЛЬНО-ІНФОРМАЦІЙНА СИСТЕМА ДІАГНОСТУВАННЯ РІВНЯ ЗНАНЬ СТУДЕНТІВ З ВИКОРИСТАННЯМ ВЕБ-ТЕХНОЛОГІЙ

© Шаховська Н.Д., Козловський Д.М., 2010

Розглянуто проблеми дистанційного навчання. Проаналізовано існуючі поширені системи тестування та перевірки знань студентів, з використанням web-технологій.

Ключові слова: тестування знань, дистанційна освіта.

This Paper concerns to distance education. The analysis is done in the common testing and verification of student education area which is based on web-technologies system.

Keywords: knowledge testing, distance education.

Вступ

Тестування є невід’ємною частиною дистанційного навчання, воно дає змогу швидко та чітко оцінити знання осіб, а обмежений час на відповіді дає впевненість в тому, що студент користуватиметься своїми власними знаннями під час розв’язання задач, не використовуючи допоміжних матеріалів. Також така система тестування являє собою зворотний зв’язок у колі “викладач” – “студент”. При цьому під час очного тестування студента викладачем виникають проблеми, пов’язані із суб’єктивністю оцінок викладачів, неможливістю для одного викладача охопити значний потік студентів, оскільки, враховуючи людський фактор, викладачу важко оцінити студентів адекватно, коли їх багато. У зв’язку з цим із розвитком інформаційних технологій особливо актуальною є проблема автоматизації процесу тестування – створення інформаційних систем тестування знань, які б не тільки забезпечували значну економію часу викладача і студента, але й давали змогу швидко і об’єктивно оцінювати знання студента, а також студенту проводити самоконтроль своїх знань з метою визначення “недоопрацьованого” матеріалу.

За останні роки спостерігається значний ріст числа навчальних закладів України, які використовують дистанційні форми навчання. Переважно вони використовують свої власні розробки програмного забезпечення. Беручи до уваги стрімке збільшення кількості користувачів і високу популярність мережі Internet, можна зробити висновок про те, що створення web-орієнтованої системи тестування знань може розширити контингент користувачів за рахунок використання єдиного для всіх користувачів мережі інструментального засобу перегляду у вигляді програми-браузера, а також загальної орієнтації на тонкого клієнта.

Актуальність роботи

Дистанційне навчання виникло відносно недавно і саме завдяки цій новизні воно орієнтується на кращий методичний досвід, накопичений різними освітніми установами в усьому світі – на використання сучасних і високоефективних педагогічних технологій, що відповідають потребам сучасного утворення і суспільства загалом. Завдяки більшій “методичній” свободі і незалежності дистанційні курси порівняно з традиційними, сформованими десятиліттями, університетським чи шкільним навчанням будуються на інноваційних підходах до навчання. Але в цьому таїться і складність – дистанційні курси, в основу яких покладено нові технології навчання, “не вписуються” у структуру і програми традиційного навчання. При сполученні подібних традиційних і інноваційних курсів їхнім розроблювачам доводиться змінювати чинні програми, проводити додаткове навчання викладачів тощо.

Також ці технології найефективніше вирішують проблеми особистісно-орієнтованого навчання. Студенти одержують реальну можливість відповідно до індивідуальних здібностей досягати визначених результатів у різних галузях знань, осмислювати одержувані знання, у результаті чого їм вдається формувати власну аргументовану точку зору на багато проблем буття.

У цьому проекті значну увагу приділяють основним компонентам дистанційного навчання, а саме: питання користувачеві для діагностування його знань та компонента тестування.

Метою роботи є проектування інтелектуальної інформаційної системи діагностування рівня знань студентів з використанням веб-технологій.

Задачами роботи є проаналізувати ринок програмного забезпечення дистанційного навчання, а також розробити програму для реалізації системи дистанційного діагностування рівня знань з використанням засобів, що надає всесвітня мережа.

Аналіз студентських відповідей є одним з найважливіших етапів перевірки знань студентів, тому в проєктованій системі цей процес буде автоматизованим, тобто за певними правилами визначатиметься результуюча оцінка. Система міститиме перелік предметів, тем та питань. Перевірка правильних відповідей визначатиметься як еквівалент еталону.

Чому цей проєкт є актуальним? Аргументи наведено нижче. Насамперед йдеться про полегшення праці викладача. Це полегшення проявляється у таких виявах:

- Зменшення затрат часу на перевірку та оцінювання робіт;
- Відсутність потреби обґрунтовувати учневі отриману оцінку (система показує, на які питання дано неправильну відповідь);
- Можливість організувати індивідуальне контролювання знань (учень має можливість самостійно здійснювати контрольні перевірки знань або дізнатись про джерела інформації для конкретної теми).

Автоматизація процесу тестування також дає змогу покращити сприйняття учнем матеріалу, а, отже, підвищити його освітній рівень. Це виявляється у такому:

- Процес самонавчання додає краще засвоїти матеріал.
- Постійна перевірка та контроль знань дають змогу показати учневі слабкі місця у вивченні матеріалу та підібрати необхідний матеріал для поглибленого вивчення окремих розділів.
- Зменшення навантаження на викладача залучають до навчання більшу кількість учнів.
- Самостійний вибір студентом навчальних дисциплін для вивчення.

Отже, задачі автоматизації процесу навчання та тестування – одні із найактуальніших задач сьогодення.

Постановка задачі та аналіз літературних джерел

За середовищем функціонування проєктована система є мережевою, тобто використовуватиметься архітектура клієнт/сервер. Обов'язковим елементом є сховище даних. Використання такого підходу забезпечує можливість централізованого контролю за роботою користувачів, а також статистичного опрацювання великої вибірки.

Існують такі підходи щодо централізованого зберігання даних на сервері, а саме:

- зберігання даних у файлах особливого формату;
- використання зовнішньої СУБД.

Проєктована система використовуватиме зовнішню СУБД. Цим уникають ряду проблем. Насамперед це швидкість роботи системи, а також можливість якісної роботи з даними. Усі сучасні СУБД мають розвинені функції архівації і відновлення даних, що в умовах посиленого обміну інформацією і необхідності доступу до даних в будь-який момент часу є доволі важливим. Особливо важливою є можливість якісної побудови різного роду статистичних характеристик. При опрацюванні літературних джерел виявлено ряд характеристик, які мають бути присутні в системі тестування знань шляхом комп'ютерного опитування [1].

- Відсутність необхідності інсталяції програми (оскільки це web проєкт).
- Незалежність від операційної системи (крос-платформеність).

- Встановлення рейтингу кожного запитання.
- Можливість встановлення рейтингу оцінок за відповідною шкалою для кожного предмета окремо.
- Можливість контролю частоти проходження тесту користувачем.
- Можливість формування відповідей за їх номерами.
- Можливість формування відповідей шляхом введення рядка.
- Можливість пропущення запитання, а з часом повернутися до нього.
- Автоматичне ведення протоколу загальних результатів : “час-студент-група-номер залікової книжки-оцінка”.
- Заборона користувачеві повторного проходження тестів швидше як за 24 год.
- Автоматичне ведення протоколу деталізованих результатів опитування за кожним студентом і кожним запитанням: “час-запитання-правильна відповідь-надана студентом відповідь”.
- Автоматичне формування результатів перевірки знань у вигляді картки з детальними результатами опитування.
- Англomовний контент для зручності проходження тестів.

Аналіз існуючих систем тестування

Moodle — безкоштовна, відкрита система дистанційного навчання. Система реалізує філософію “педагогіки соціального конструктивізму” та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання. Moodle перекладена десятками мов, зокрема існує частковий переклад українською. Система використовується у 175 країнах світу, а також і в Україні. Національний університет “Львівська політехніка” широко використовує цю систему для оцінювання студентів. Також потрібно зазначити, що цей проект є відкритим та в ньому бере участь велика кількість інших розробників [8].

Moodle написана PHP з використанням SQL-бази даних. Moodle може працювати з об'єктами SCO та відповідає стандарту SCORM.

ATutor є системою керування навчальним матеріалом (Learning Content Management System, LCMS). Програма є простою у встановленні, налаштуванні та підтримці для системних адміністраторів; викладачі (інструктори) можуть доволі легко створювати та переносити навчальні матеріали та запускати свої онлайн-курси. А оскільки система є модульна, тобто складається з окремих функціональних одиниць – модулів, то вона відкрита для модернізації і розширення функціональних можливостей.

Claroline є спільною електронною навчальною разом з і eWorking платформою (Learning Management System) випущеною під ліцензією Open Source. Це дає змогу сотням організацій по всьому світу (університети, школи, компанії, асоціації, ...) створювати й адмініструвати курси та співпрацювати через Інтернет. Платформу використовують у понад 80 країнах більш ніж 30 мовами.

eCollege – це програмне забезпечення як послуга (SaaS), постачальник електронного навчання програмного забезпечення і послуг для середнього та вищого навчальних закладів. Заснована у 1996 році в Real освіти, компанія стала публічною у 2002 році eCollege.com (NASDAQ: ECLG).

Діаграма прецедентів

Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Прецеденти є основним засобом визначення необхідної поведінки системи. Як правило, вони використовуються для описання вимог до системи, тобто що має робити система. Основними поняттями, пов'язаними з прецедентами в розробленій діаграмі, є актори – такі, як: Користувач та Сервер БД актори, прецеденти, такі як: Реєстрація, Логування, Тестування, та суб'єкт. Суб'єкт — це система, що розглядається і до якої належать прецеденти – тобто Система

діагностування знань. Користувачі та будь-які інші системи, що можуть взаємодіяти із суб'єктом, представлено як акторів. Кожну систему можна описати з різних боків – зокрема ось представлення системи діагностування знань.



Рис1. Діаграма прецедентів

Основний матеріал

Розробляючи автоматизовану систему дистанційної перевірки і оцінювання, використано архітектуру клієнт/сервер. Для проектування цієї роботи потрібно використати web-сервер з підтримкою технології CGI. Тестова система міститиме інформацію про групи студентів, самих студентів, предмети, тести з цього предмета, тестові завдання і відповіді, а також результати проведення контролю знань. Для цього використано систему управління базами даних (СУБД) MySQL.

Однією з головних переваг цієї системи є її багаторівневність. Питання є поділені за складністю та за типами саме так, що користувач проходить спочатку перевірку знань на середньому рівні, і якщо його результат є добрим, переходить на другий рівень, в інакшому ж випадку він потрапить на нижчий рівень, де йому доведеться відповідати на простіші запитання, які містять менше балів, саме для того, щоб детально, достовірно та з мінімальною похибкою оцінити рівень знань користувача. Логіка переходу буде задаватися простим оператором if, який, своєю чергою, буде викликатись, якщо сторінка-обробник буде видавати поточні результати користувача. Логіка є такою.

Якщо користувач набере понад 75 балів на першому рівні оцінювання знань, його тест буде завершеним, і оцінку буде виведена на екран.

Якщо користувач набере понад 50 балів, але менше або рівно 75 балів на першому рівні оцінювання знань, його тест буде продовжений на вищому рівні для детальнішого оцінювання знань.

Якщо користувач набере менше 50 балів, його тестування буде продовжене на нижчому рівні для детальнішого оцінювання його знань.

Отже, з логікою переходу на рівні тестування ми ознайомилися, тепер повернемося до СУБД. Зобразимо архітектуру системи.

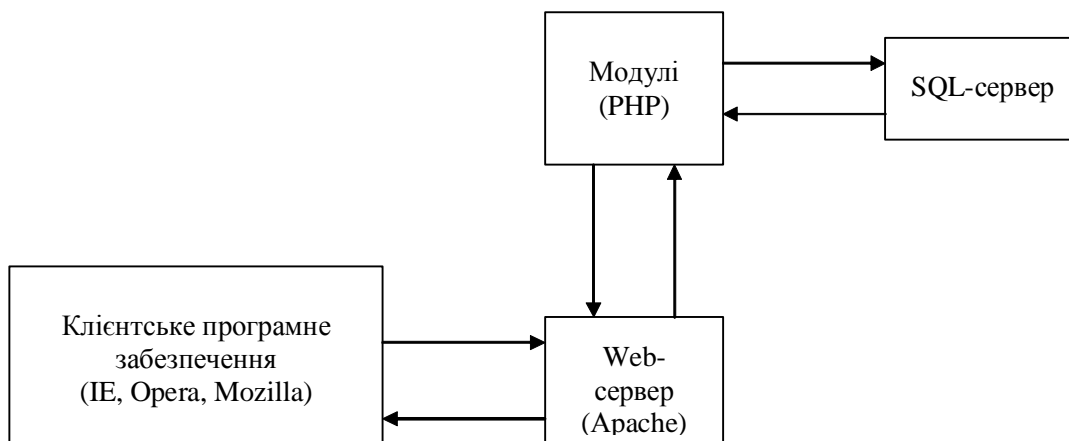


Рис. 2. Архітектура компонентів системи

Як показано на рис. 2, як web-сервер використовується Apache, під управлінням якого працює понад 60 % всіх відомих вузлів [4].

Опишемо принцип функціонування системи. Користувач за допомогою програми навігатора по мережі Інтернет задає URL-адресу сервера, який йому необхідний. Клієнтська програма шукає відповідний сервер за заданою URL-адресою, а знайшовши, передає йому управління. У конфігураційному файлі web-сервера є вказівка, яка сторінка повинна завантажуватись першою, тобто, якщо користувач введе URL-адресу, наприклад: <http://www.borland.com>, web-сервер повинен завантажити початкову сторінку. У розробленій тестовій системі використовується програма web-сервер Apache, в його конфігураційному файлі є рядок, який має вигляд [4]:

```
DirectoryIndex index.htm index.html index.shtml
```

Цей рядок вказує web-серверу, що за замовчуванням початкова сторінка знаходиться у файлі index.htm, index.html або index.shtml. Якщо взяти для прикладу web-вузол <http://www.borland.com> та конфігураційний файл web-сервера з таким рядком, то повна адреса матиме вигляд: <http://www.borland.com/index.htm>.

SQL-сервер – це система управління базами даних, яка розташовується на сервері. На сервері також зберігаються бази даних, з якими працює SQL-сервер. У розробленій системі як SQL-сервер використовується система управління розподіленими базами даних – MySQL.

За результатами аналізу продуктивності даної СУРБД [6] вона вважається однією із найшвидших у роботі, правда це досягається за рахунок того, що у цьому сервері відсутні деякі можливості, які дуже значно впливають на продуктивність роботи системи [5].

Для того, щоб можна було використовувати інформацію з бази даних тестової системи, необхідно їх звідти якимось чином викликати. Для цього використовується технологія CGI. Користувач викликає за допомогою клієнтської програми сервер тестової системи, на початковій сторінці якого є форма, де користувач вводить свої персональні дані, за якими він ідентифікується в системі.

Тестування за формою представлення інформації

Одним із основних блоків спеціалізованого математичного забезпечення автоматизованих навчальних систем (АНС) є блок аналізу відповідей користувача, який функціонує так. При використанні будь-якого способу вводу відповіді виділяються деякі ознаки, відповідно до яких відповідь належить до категорії правильних або неправильних [2].

Деякі системи контролю знань підтримують [1] також окремі специфічні форми відповідей, наприклад, дають змогу створювати структурні схеми на заданій елементній базі або вибирати

визначену область нїа графічному зображенні. Але такі форми відповідей не мають широкого застосування, оскільки не володіють достатнім рівнем абстракції. У тестовій системі буде введена можливість перегляду сесії користувача, яка дасть змогу переглянути, коли саме, як саме результативно та з якого саме предмета проходив користувач тестування. Цей функціонал дасть змогу оцінювати знання користувача, їх порівнювати та робити висновки про успіхи студента, що є дуже важливим аспектом у навчанні.

На основі аналізу чинних тестових систем визначено такі характеристики тестової бази проектованої системи:

– Типи завдань (за формою представлення):

- а) Текст;
- б) Зображення;
- в) Процес.

– Типи відповідей (за формою вводу і представлення):

- а) Вибірка елементів (невпорядкована);
- б) Список елементів (впорядкований);
- в) Вираз (арифметичний);
- г) Фраза (текст);
- д) Зображення.

Тестування знань студентів повинно бути специфікованим до певних тем та областей. У цьому проекті тестування буде поділене на три категорії, а також кожна категорія буде поділена, своєю чергою, на підкатегорії. Отже, наступні категорії тестування будуть доступні в проектованій системі:

- 1. English Grammar
 - 1.1. Simple Present or Present Continuous;
 - 1.2. Simple Past or Past Continuous;
 - 1.3. Present Perfect or Simple Past.
- 2. Programing (base knowledge)
 - 2.1. Java script;
 - 2.2. C;
 - 2.3. XML.
- 3. HTML & DataBase & QA
 - 3.1. SQL transactions;
 - 3.2. HTML;
 - 3.3. QA.

Основним етапом в процесі оцінювання знань є саме перевірка відповідей на правильність. У випадку онлайн-системи діагностування знань потрібно зробити можливість передачі даних правильної відповіді з інтернет сторінки на сторону сервера, а також разом з цим захистити нашу систему від того, щоб користувач цього не бачив. Можна використати декілька підходів, але цей проект буде реалізований за допомогою одного з полів HTML, яке називається прихованим (hidden field).

Приховане поле HTML

Приховане поле не показується на сторінці і приховує свій вміст від користувача. Відвідувач не може нічого в нього внести або надрукувати. Мета створення прихованих полів – це є передача технічної інформації на сервер. У більшості випадків це необхідно для передавання даних форми від сторінки до сторінки. `Type=hidden` Створює прихований елемент, що не відображається користувачеві. Інформація, що зберігається в прихованому полі, завжди пересилається на сервер і не може бути змінена ні користувачем, ні браузером. Приховане поле можна використовувати, наприклад, у такому випадку. Користувач заповнює форму і відправляє її серверу. Сервер посилає користувачеві для заповнення другу форму, яка частково використовує інформацію, що міститься в

першій формі. Сервер не зберігає історію діалогу з користувачем. Він обробляє кожен запит незалежно і при отриманні другої форми не знатиме, як вона пов'язана з першою. Щоб повторно не вводити вже введену інформацію, можна змусити Sgi-програму, оброблювальну першу форму, переносити необхідні дані в приховані поля другої форми. Вони не будуть видимі користувачем і водночас будуть доступні серверу. Значення прихованого поля визначається атрибутом VALUE (табл. 1).

Синтаксис створення прихованого поля.

`<input type="hidden" параметри>`

Таблиця 1

Параметри прихованого поля

Параметр	Опис
name	Ім'я поля для його ідентифікації обробником форми.
value	Значення поля, яке визначає, яка інформація буде відправлена на сервер.

Приклад використання прихованих полів наведено нижче

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<title>Приховане поле</title>
</head>
<body>
<form action="#" method="post">
<p><b>Напишіть улюблене слово (ніякі дані не будуть передаватись на сервер!</b></p>
<p><input type="text" size="25" name="word">
<input type="hidden" name="UserName" value="Bill">
<input type="hidden" name="password" value="Un123"></p>
<p><input type="submit" value="Відправити"></p>
</form>
</body>
</html>
```

У цьому прикладі показано створення двох прихованих полів, одне з яких має назву UserName і отримує значення Bill, а друге іменується password зі значенням Un123. У результаті відправки форми, зазначеної в параметрі action оброблювачеву, програма може легко прочитати ці дані й інтерпретувати їх на розсуд розробника.

Звичайно, що на одному прихованому полі доброї тестової системи не зробити, отож, у нашій системі буде передбачено декілька правильних відповідей, а якщо їхня форма представлення є "текстове поле", то ми змушені використовувати регулярні вирази (RegExp) для аналізу відповідей.

Регулярні вирази у тестовій системі

Регулярний вислів — це рядок що описує або збігається з множиною рядків, відповідно до набору спеціальних синтаксичних правил. Регулярні вислови використовуються в багатьох текстових редакторах та допоміжних інструментах для пошуку та зміни тексту на основі заданих шаблонів. Багато мов програмування підтримують регулярні вислови для роботи з рядками. Наприклад, Perl має потужний механізм для роботи з регулярними висловами, вбудований

безпосередньо в його синтаксис. Завдяки набору утиліт (включаючи редактор *sed* та фільтр *grep*), що входили до складу дистрибутивів Юнікс, регулярні вислови стали відомими та поширеними. У тестовій системі широко будуть використовуватись регулярні вирази, тому для детальнішого ознайомлення з ними є таблиці (табл. 2).

Таблиця 2

Квантифікатори

Позначення	Число повторів
*	Ноль або більше
+	Одно або більше
?	Ноль або один раз
{n}	Точно n відповідей
{n,}	n або більше відповідей
{n,m}	Мінімум n та максимум m відповідей

Саме за допомогою квантифікаторів ми визначатимемо правильну кількість повторень. У PHP можливі два типи регулярних виразів: **Perl-сумісні** (сучасні, які використовуються функціями *preg_match*, *preg_replace* тощо) і **POSIX** (застарілі, використовуються функціями *ereg*, *ereg_replace* тощо). Розберемо одну з найчастіше використовуваних функцій: *preg_match* (*string \$pattern* , *string \$subject* [, *array \$matches*]) шукає в заданому тексті *subject* збіг з шаблоном *pattern*. У випадку, якщо додатковий параметр *matches* вказано, він буде заповнений результатами пошуку. Елемент *\$matches[0]* міститиме частину рядка, відповідну входженню всього шаблону. *\$matches[1]* – частину рядка з першої підмаски і так далі. А за допомогою наступних виразів ми визначимо правильність відповідей (якщо вони двохзначні) в текстових полях (табл. 3).

Таблиця 3

Позиція в рядку

Представлення	Позиція
^	Початок рядку
\$	Кінець рядку
\b	Закінчення слова
\B	“Не закінчення” слова

Засоби реалізації

Цю систему керування базами даних (СКБД) з відкритим кодом було створено як альтернативу комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL— одна з найпоширеніших систем керування базами даних. Вона використовується насамперед для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Підключення кожної мови характеризується своїм власним інтерфейсом, який визначає правила доступу до баз даних СУБД MySQL. Зупинимось на описі однієї з найпопулярніших [5].

PHP – це скриптова мова програмування, яка була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважно більшістю хостинг-провайдерів. PHP— проект відкритого програмного забезпечення. PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта.

На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з погляду безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів, які виконуються вже на стороні клієнта. Наявність інтерфейсів до багатьох баз даних робить PHP гнучкою мовою

програмування. В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.

Для розроблення системи використано PHP API. PHP є мовою для написання сценаріїв, яка є зручною для включення програм в тіло Web-сторінки. Така сторінка обробляється PHP перед тим, як буде відіслана клієнту, що дає змогу таким сценаріям генерувати динамічні Web-сторінки, які містять результат SQL-запиту до бази даних [6].

У процесі вибору типу інтерфейсу API враховували багато факторів, а саме:

- Середовище, в якому виконуватиметься задача. Охарактеризуємо його як контекст, в якому виконуватиметься програма.

- Продуктивність. Тобто ефективність програм, написаних із застосуванням того чи іншого інтерфейсу.

- Простота розробки: наскільки зручно працювати з інтерфейсом і його мовою при створенні програм.

- Переносимість: чи будуть використовуватися програми для роботи з іншими базами даних.

Мову PHP розробляли як інструмент розробки Web-програм, тому вона є середовищем, в якому її використання буде найправильнішим. Водночас, доступ до бази даних є найбільшим привілеєм. Отже, цілком зрозуміло, її необхідно використовувати в програмах, які здійснюють доступ до бази даних через Internet. Звичайно, можна використовувати PHP і в ролі автономного інтерпретатора. Мови Perl і PHP самі здійснюють управління пам'яттю, і ймовірність їх краху в результаті помилок, викликаних управлінням пам'яттю, значно менша, тому проєктована система виконуватиметься саме завдяки розробленню на PHP [7].

Аналіз необхідного програмного забезпечення

Для написання програмного коду автоматизованої системи використовують таке програмне забезпечення:

- Apache Web Server;
- PHP Module;
- MySQL Server and Clients;
- Zend Studio (ZDE);
- MySQL Turbo Manager;
- Sybase Power Designer.

Висновки

Розглянуто чинні системи діагностування знань, визначено цілі, засоби та методи імплементації проєкту.

Здійснено аналіз процесу опитування та процесу визначення рівня знань користувача. Розглянуто структуру процесу оцінювання альтернативних варіантів. Розглянуто основні компоненти системи.

1. <http://www.distance-learning.ru/db/el/F4AF5863268B4F4AC3256DCC0045944B/>. 2. Засоби дистанційного навчання. Методика, технологія, інструментарій / С.В. Агапонов, З.О. Джалиашивили та ін.. – СПб., 2003. 3. Синепол В.С., Цикин И.А. Применение современных программно-аппаратных средств компьютерной видеоконференц-связи для создания интерактивных информационно-обучающих систем / Мат-лы Всероссийской научно-методической конф. “Телематика 95”. – СПб., 1995 4. Никитин А.Б., Синепол В.С., Сороцкий В.А., Цикин И.А. Интерактивные информационные технологии на основе Web-серверов и систем компьютерной видеоконференц-связи // Дистанционное образование, 1998, №1. 5. Боуман Дж.С., Эмерсон С.Л. Практическое руководство по SQL; 352 стр. 6. Полякова Л.Н. Издательство: Интернет-университет информационных технологий, 2004; 368 с. 7. Вайк, Уокер, Кокс PHP (справочник) — ДиаСофт 2001 448 стр. 8. <http://www.moodle.com>.