

Міністерство освіти і науки України
Луцький національний технічний університет

В. О. Сацик
А. І. Гарлінський
Д. О. Сомов

Універсальна система трансформації даних: медприлади – web-інтерфейс

Монографія

Луцьк
Вежа-Друк
2014

ББК 73.068
УДК 681.518
С 41

*Рекомендовано до друку вченою радою
Луцького національного технічного університету
(протокол № 10 від 20.06.2014 р.)*

Рецензенти:

Швабюк В. І. – доктор технічних наук, професор (Луцький національний технічний університет);

Марчук В. І. – доктор технічних наук, професор (Луцький національний технічний університет);

Тулашвілі Ю. Й. – доктор педагогічних наук, професор (Національний університет водного господарства та природокористування).

Сазик В. О.

С 41 Універсальна система трансформації даних: медприлади – web-інтерфейс : монографія / В. О. Сазик, А. І. Гарлінський, Д. О. Сомов. – Луцьк : Вежа-Друк, 2014. – 120 с. : ілюстр.

У монографії представлено результати досліджень процесів трансформації даних параметрів здоров'я людини (отримуваних від різнотипових локальних медприладів) в он-лайн середовище на основі розроблюваних модулів із розширеним набором індивідуально створених кастомних функцій, використавши таксонометричні словники й тип матеріалу, які реалізовані в CMS Drupal.

Написано унікальний код, що є досить гнучким і може бути легко впроваджений в інші зовнішні системи; забезпечує мінімальне навантаження на сервер, що важливо при обробці великої кількості даних і безперервного потоку користувачів на сайт та мобільний додаток.

Для фахівців у галузі інформаційних, медичних технологій та установ.

**ББК 73.068
УДК 681.518**

© Сазик В. О., Гарлінський А. І.,
Сомов Д. О., 2014
© Маліневська І. П. (обкладинка), 2014

ЗМІСТ

ст.

ВСТУП.....	6
РОЗДІЛ 1 ВИЗНАЧЕННЯ ПОНЯТЬ СИСТЕМА, МЕДПРИЛАДИ, ТРАНСФОРМАЦІЯ; ОБГРУНТУВАННЯ ВИБОРУ СУБД MYSQL ТА СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ CMS DRUPAL (Огляд літературних джерел).....	13
1.1 Визначення поняття «система».....	13
1.2. Поняття «медприладів»	20
1.3 Поняття «трансформація даних».....	22
1.4 Визначення поняття «База даних» (БД).....	23
1.5 Класифікація СУБД за типом даних керування	26
1.6 Система управління контентом WordPress.....	29
1.7 Система управління контентом Joomla.....	30
1.8 Система управління контентом Drupal.....	32
Висновки.....	37
Список використаної літератури.....	39
РОЗДІЛ 2 СТРУКТУРА БАЗИ ТА ТИПИ ПОЛІВ СУБД MYSQL.....	40
2.1 Архітектура СУБД MySQL.....	40
2.2 Системи забезпечення зв'язку СУБД MySQL з програмними продуктами	43
2.3 Зв'язок PHP з MySQL.....	46
2.4 Система зв'язку з СУБД MySQL (phpMyAdmin).....	47
Висновки.....	48
Список використаної літератури.....	49
РОЗДІЛ 3 ОПИС ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «ЗДОРОВ'Я».....	50
3.1 Призначення, будова, технічні характеристики, принцип роботи діагностичного пристрою «ЗДОРОВ'Я».....	50
3.2 Аналіз результатів параметрів стану здоров'я людини.....	57
Висновки.....	58
Список використаної літератури.....	59
РОЗДІЛ 4 АЛГОРИТМ РОБОТИ СИСТЕМИ ОТРИМАННЯ І ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ З ІНТЕРЕНТ ІНТЕРФЕЙСОМ НА СУБД MySQL.....	60
4.1 Огляд баз даних для створення алгоритму.....	60
4.2 Завантаження архіву від програмного забезпечення мед-приладів з XML файлами.....	63
4.3 Розархівування архіву та отримання його вмісту (XML файли	65
4.4 Передача даних із XML файлів у PHP.....	68
4.5 Обробка XML файлів, перетворення вмісту у масив даних системи.....	70

4.6 Запис батча та обхід масивна по елементно (парсинг) за допомогою розробленого функціоналу.....	71
4.7 Трансформація даних з програмного забезпечення медприладу до заданного вигляду у системі використовуючи таксонометричні словники....	72
4.8 Збереження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL.....	72
4.9 Аналіз та трансформація даних.....	73
4.10 Створення користувачів	73
4.11 Створення «візитів».....	73
4.12 Генерація інтернет - інтерфейсу: графіки візиту, графіки динаміки, рекомендації.....	74
4.13 Мобільний Android додаток	75
Висновки	75
Список використаної літератури.....	76
РОЗДІЛ 5 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ РОБОТИ СИСТЕМИ ОТРИМАННЯ І ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ З ІНТЕРНЕТ ІНТЕРФЕЙСОМ НА СУБД MySQL.....	77
5.1 Завантаження архіву з XML файлами від програмного забезпечення медприладів.....	77
5.2 Розархівування архіву та отримання його вмісту (XML файли).....	79
5.3 Отримання вмісту архіву.....	79
5.4 Обробка XML файлів, перетворення їх вмісту у масив даних системи....	81
5.5 Створення черги.....	81
5.6 Запуск батча	82
5.7 Парсинг даних.....	83
5.8 Трансформація даних.....	84
5.9 Завантаження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL таблиця NODE та визначення стійкості роботи системи.....	85
Висновки	86
Список використаної літератури.....	86
РОЗДІЛ 6 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОЇ СИСТЕМИ ОТРИМАННЯ І ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ З ІНТЕРНЕТ ІНТЕРФЕЙСОМ НА СУБД MYSQL.....	87
6.1 Вигляд даних у програмному забезпеченні апарату діагностичного пристрою «ЗДОРОВ'Я».....	87

6.2 Вигляд даних у системі з інтернет-інтерфейсом на СУБД MYSQL.....	88
6.3 Візуалізація даних.....	89
6.4 Приклад синтезу PHP та JavaScript коду який виводить середньо – статистичні висновки про стан енергетичних меридіанів у вигляді графіків.	91
6.5 Приклади створеного інтерфейсу системи отримання і збереження інформації на СУБД MySQL.....	94
6.6 Обґрунтування необхідності створення мобільного додатку.....	101
6.7 Середовище розробки.....	102
6.8 Мова програмування мобільного додатку.....	102
6.9 XML розмітка.....	103
6.10 JAVA код.....	104
Висновки	106
Список використаної літератури	107
РОЗДІЛ 7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБКИ СИСТЕМИ ТРАНСФОРМАЦІЇ ДАНИХ ВІД МЕДПРИЛАДІВ У WEB –ІНТЕРФЕЙС	108
Загальні висновки	112

ВСТУП

Рівень медицини Ізраїлю, Германії, США, Швейцарії, Бельгії та ін.. країн Європи на порядок вище ніж в країнах СНД і в Україні зокрема. Основними елементами високоякісної медицини в даних країнах є:

- кваліфікований персонал;
- високий рівень сервісного обслуговування;
- проведення та впровадження новітніх досліджень;
- застосування новітніх обладнання та технологій (комп'ютерних зокрема);
- поступовий перехід медичних установ даних країн в інформаційний режим;
- тісний зв'язок з науковими установами та використання їх наробок.

Зокрема в Ізраїлі, паралельно з наданням медичної допомоги ведеться багатогранна та інтенсивна наукова робота. Одним з найбільших і високотехнологічних медичних центрів Ізраїлю є багатопрофільний Медичний центр імені Рабіна (МЦР) і дитячий медичний центр Шнайдер. МЦР є клінічною базою Вищої медичної школи ім. Саклера при Тель-Авівському університеті, підтримує наукові контакти з провідними медичними установами світу (http://dobrobut.com/rabin_medical_center.html).

Відповідно передові клініки нашої країни намагаються наслідувати і в якійсь мірі відповідати рівню зарубіжній медицині. Принаймні закупівлею та використанням сучасної техніки діагностування.

Але не варто забувати про те, що велика кількість досліджень і нововведень використовуваних закордоном є напрацюванням наших з вами співвітчизників.

Напрошується висновок: навіщо використовувати і впроваджувати закордонні технології, якщо ми спроможні і повинні зробити все для того, щоб відповідати високим медичним стандартам.

Першим кроком який ми могли б зробити і впровадити, це перейти на цифровий вигляд карт пацієнтів, запис пацієнта на прийом в онлайн режимі,

результатів їх візитів, рекомендацій, по встановленому діагнозу, призначення лікування і по можливості впровадження онлайн консультацій,. Загалом це говорить про необхідність створення веб-інтерфейсу.

Що гарантовано забезпечить медичному закладу та пацієнту створення веб-інтерфейсу:

- зменшення кількості черг і людей у черзі;
- задіювання в одному ресурсі великої кількості лікарів з можливістю створення кімнати спілкування за окремим (важким) випадком і отримування консолідованої думки інших лікарів у даній проблемі, де вся історія пацієнта може бути синхронно переглянута кожним з учасників в онлайн режимі.
- можливість вибору пацієнтом онлайн в режимі відділення для консультації;
- запис на прийом до лікаря в онлайн режимі.
- збереження та доступ в онлайн режимі (в інтернет інтерфейсі, додатку на мобільному телефоні) до інформації як зі сторони лікаря так і пацієнта з можливістю консультацій в іншій медичній установі.

Плюси які дає такий перехід можна називати дуже довго, багато з них можна спростовувати, але на те ми маємо розум і руки щоб приходити до спільного рішення і допрацьовувати наші нововведення для покращення нашого з вами здоров'я та продовження життя.

Актуальність роботи

Провівши аналіз досліджень у сфері здоров'я людини можна зробити висновок, що це направлення цікавить людство чи не найбільше. В останній час, з появою можливості доступу до інтернету практично кожному, доступності технічних пристроїв, з'являється неймовірна кількість додатків та систем слідкування за здоров'ям людини. Такими є: щоденники артеріального тиску, додатки з систем щоденного тренування і т.д.

Не дивлячись на чималу кількість представлених розробок котрі забезпечують он-лайн режим слідкування за окремими показниками здоров'я, вони потребують певного удосконалення, для того, щоб забезпечити зростаючі потреби пацієнтів та лікарів.

Оперативному прослідковуванню в он-лайн режимі динаміки змін показників стану здоров'я в цілому як зі сторони пацієнтів так і зі сторони лікарів, формуванню вибраним лікарем оптимальної медичної рекомендації, в он-лайн режимі без повторного обстеження, присвячена дана робота.

Мета та завдання

Мета роботи: розробити універсальну систему трансформації даних від мед-приладів (локальний рівень, графічно-текстовий вигляд) в web-інтерфес (глобальна інтрнет-мережа, з можливістю відображення в додатках) на базі програмного продукту розробленого на CMS Drupal, та додатку використовуючи платформу Android.

Мета роботи буде реалізована на основі вирішення наступних низки **завдань:**

- розкрити поняття система, медичні прилади, трансформація
- обґрунтувати вибір СУБД MYSQL та систем управління контентом CMS DRUPAL;
- написати алгоритм роботи універсальної системи трансформації даних від мед-приладів у web-інтрефес для цього необхідно:
 - створити веб-форму для завантаження архіву з XML файлами;
 - забезпечити отримання вмісту архіву;
 - сформувати передачу даних із XML файлів у PHP
- створити PHP аналізатори для обробки вхідної інформації;
- реалізувати збереження даних у БД MySQL;
- забезпечити посилання запитів у БД та отримання відповіді;

- розробити функцію обробки отриманої відповіді та відображення її у вигляді стовпчастих графіків;
- обґрунтувати та в разі необхідності створити мобільний додаток;
- провести економічну оцінку розробленої системи.

Об'єкт дослідження.

Процес трансформації даних параметрів здоров'я людини (отримуваних від локальних медприладів) у он-лайн середовище завдяки розробленим нами модулів (з розширеним набором індивідуально створених кастомних функцій) використавши таксонометричні словники та тип матеріалу які реалізовані в CMS Drupal.

Процес трансформації буде реалізовано трьома основними етапами:

- вилучення інформації з джерела даних (локальний медприлад) та визначення складу даних;
- очистка та перетворення (структурування) даних у формат створюваної системи;
- відображення структурованих даних в веб-інтерфейсі та додатку (по запиту).

Предмет дослідження.

Програмно-апаратний комплекс «Здоров'я», система отримання та збереження інформації з інтернет інтрефейсом розробленої на CMS Drupal за допомогою СУБД MySQL, Android - мобільний додаток.

Методи дослідження.

Поставлена мета та задачі дослідження вирішувались методом аналізу та синтезу, логікою форми і змісту.

Методом аналізу форми отримано ієрархічну архітектуру даного проекту (рукопис машинописного тексту, веб-інтерфейс, мобільний додаток).

Методом аналізу змісту ми визначили мету та завдання нашої роботи; набір оптимальних складових для розробки створюваної системи.

Методом синтезу форми було отримано алгоритм та його програмна реалізація роботи системи отримання та збереження інформації з веб-інтерфейсом на СУБД MySQL.

Методом синтезу змісту було отримано процес трансформації даних параметрів здоров'я людини (отриманих на основі «Програмно-апаратного комплексу Здоров'я») в он-лайн середовище.

Використовувана нами методологія дослідження включає в себе також і класичні наукові підходи: розрахунково-аналітичний метод (побудова графіків), метод процедурного програмування (написання програмного коду), метод моделювання (імітація отримання інформації з зовнішніх систем за допомогою розробленої форми у веб-інтерфейсі), метод динамічного програмування (написання гнучкого, пластичного програмного коду до кількісних та якісних змін вхідних даних).

Наукова новизна

Розробка універсальної системи котра забезпечуватиме процес трансформації даних від різного типу медприладів у web-інтерфес, в тому числі із можливістю відображення на мобільних додатків.

Проведено практичне тестування розробленої системи на основі «Програмно-апаратного комплексу Здоров'я».

Рішення даної задачі вдалося реалізувати на основі розробки власних унікальних обробників:

- batch: мінімізації навантаження серверної частини, використовуючи метод вибудови черги обходу елементів масиву даних створенням програмного пластичного коду на основі науково обгрунтованого аналізу можливих вхідних даних та синтезу успішно функціонуючих обробників в інших галузях;
- parser: обхід масива поелементно під специфіку вхідних даних;
- transformation: унікальний обробник перетворення специфічних вхідних даних у заздалегідь визначений нами вигляд з використанням таксонометричних словників, а разі збільшення (нововведення) кількості та якості вхідних параметрів вони автоматично будуть сприйматися, зберігатися та структуровуватися системою.

Стійкість та універсальність створеної системи отримання і збереження інформації з веб-інтерфейсом на СУБД MySQL вирішено також за рахунок:

- інваріантності властивості системи до відхилення від заданих початкових (вхідних) параметрів, яка забезпечується універсальністю обробників;
- створення універсальних обробників (на основі адаптивності коду до зміни вхідних даних) архіву з XML файлами (вхідних даних) з подальшим перетворенням у вигляд доступний користувачу в веб-інтерфейсі;

Створено веб-інтерфейс гарантуючий отримання:

- комфортного формування та відображення електронної медичної картки в зручній для користувача формі про стан систем організму кожного візиту, їх динаміки.
- зберігання даних про пацієнтів їх візитів, параметрів здоров'я та формування електронної медичної картки.

Практичне значення одержаних результатів

1) Можливість ефективного ведення електронної медичної картки за отриманими параметрами здоров'я, динаміки їх зміни, сформованої медичної рекомендації з оперативним доступом незалежно від розміщення в просторі та часі (за наявності доступу до інтернету).

2) Оперативне забезпечення лікаря змістовною інформацією про стан параметрів складових систем організму у вигляді стовпчатих графіків, їх еталонної шкали, впливу на функціонування організму в цілому та формування оптимальної рекомендації пацієнту.

3) Поширення систем збереження, обробки, та виводу інформації про стан здоров'я людини на електронних носіях та інтернет середовищі.

4) Підвищення рівня культури обслуговування у медичних закладах.

5) Забезпечення функціональної сумісності розробленої системи на етапах запуску та доводки з будь якими цифровими медичними апаратами.

Автори висловлюють щире подяку Разкевичу Петру Васильовичу директору фірми LTD "Iron Eggs" та доценту кафедри АУВП Луцького НТУ Лотишу Володимирі В'ячеславовичу за надання консультаційної допомоги в процесі розробки універсальної системи трансформації даних від медприладів у web-інтрефес.

Публікації по роботі

1) В.О. Сазик, О. М. Решетило, О.В. Сазик Обґрунтування вибору систем керування базою даних та управління контентом при створенні Web-сайту // Міжвузівський збірник «Комп'ютерно-інтегровані технології: освіта, наука, виробництво», 2013. - №12. - С.56-62.

2) В.О. Сазик, Л.М. Маркіна, Д.О. Сомов

Алгоритм роботи системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL // Міжвузівський збірник «Комп'ютерно-інтегровані технології: освіта, наука, виробництво», 2014. - №12. - С.56-62.

Обсяг і структура роботи

Монографія складається із вступу, семи розділів, висновків та пропозицій виробництву, списку використаних джерел які включають. Робота викладена на 120 сторінках машинописного тексту, містить 6 таблиць, 11 рисунків, 1 фотографію та 3 схеми.

РОЗДІЛ 1

ВИЗНАЧЕННЯ ПОНЯТЬ СИСТЕМА, МЕДПРИЛАДИ, ТРАНСФОРМАЦІЯ; ОБГРУНТУВАННЯ ВИБОРУ СУБД MYSQL ТА СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ CMS DRUPAL

(Огляд літературних джерел)

1.1 Визначення поняття «система»

В даний час немає єдності у визначенні поняття "система", в підходах до класифікації систем, в трактуванні основних системних закономірностей. Наукові напрямки, які спираються на системні уявлення, носять в літературі різні назви. Мета даного розділу – забезпечити орієнтування в різних тлумаченнях щодо систем, які допоможуть у вирішенні практичних завдань, котрі виникають при вивченні питань розробки та організації функціонування складних людино-машинних комплексів і систем управління. З цією метою в проведемо аналіз визначення поняття "система"; розглянемо еволюцію цього поняття; ознайомимося із основними властивостями систем.

Визначення поняття система змінювалося не тільки за формою, а й за змістом.

Так, в перших визначеннях у тій чи іншій формі говорилося про те, що система - це сукупність елементів і зв'язків (відносин) між ними. Наприклад, основоположник теорії систем Л. фон Берталанфі визначав систему як комплекс взаємодіючих елементів, або як сукупність елементів, що перебувають у певних відносинах один з одним і з середовищем. А. Холл визначає систему як безліч предметів разом зі зв'язками між предметами і між їхніми ознаками. До сих пір ведуться дискусії, який термін - "ставлення" або "зв'язок" - краще вживати [2].

І все ж, у більшості випадків прагнуть не тільки включити у визначення системи поняття елементу та зв'язку (або відносини), а й уточнити хоча б одне з

них. Для цього в визначення включають властивості. У наведеному визначенні А. Холла властивості (ознаки, атрибути) доповнюють поняття елемента, предмета. У визначенні А. И. Уйомов властивості можуть характеризувати як елементи, так і відносини. Уточнюючи властивості відносин, Ю. А. Урманцев виділяє у визначенні системи відносини між елементами і закони композиції [1].

Потім у визначеннях з'являється поняття мети. Спочатку в неявному вигляді. У ряді визначень поняття мети як би включається в поняття цілісності. Так, в "Філософському словнику" система визначається як «сукупність елементів, що перебувають у взаємних зв'язках співвідпорядкування й координації і складають певну цілісність» [4].

Ще згодом, мета у визначеннях з'являється в більш явному вигляді, тобто або у вигляді призначення системи, або кінцевого результату, або системоутворюючого критерію і т. п. (це визначення В. І. Вернадського, У. Р. Гібсона, П. П. Анохіна) або навіть з явним згадуванням про мету (визначення В. Н. Сагатовського).

Далі у визначення поняття системи, особливо останнім часом, починають включати, поряд з елементами, зв'язками їх властивостями і цілями, - спостерігача, особа, що представляє об'єкт або процес у вигляді системи. Слід зазначити, що вперше на необхідність врахування взаємодії між дослідником і досліджуваною системою вказав У. Р. Ешбі.

Українська Радянська Енциклопедія дає наступне визначення поняттю система: (від грец. – поєднання, устрій, утворення) – сукупність якісно визначених елементів (зміст системи), між якими існує закономірний зв'язок чи взаємодія (структура системи) [3].

Систему можна також представити у вигляді деяких графічних структур з не до кінця виявленими елементами і зв'язками або на її уявлення використовувати "мову", який може бути заснований на теоретико-множинних, логічних та інших формалізованих уявленнях.

Аналізуючи різні поняття системи та їх еволюцію і не виділяючи поки, що жодне з них в якості основного, ми хотіли не лише показати складність короткого визначення таких зазвичай інтуїтивно осягаються понять, як система, а й дати зрозуміти, що на різних етапах при поданні різних аспектів однієї і тієї ж досліджуваної або конструйованої системи можливі, різні визначення (а якесь визначення завжди потрібно прийняти як "робоче"); більш того, при переході до наступного етапу прийняте раніше визначення може бути замінено іншим.

Читач, напевно вже задав запитання, «чому для поняття системи приділено так багато уваги»? Річ в тім, що від правильності сприйняття того чи іншого поняття буде залежати ефективність самої розробки (системи) та якість її функціонування.

В нашому розумінні найбільш правильно і смислоємко поняття система визначає в своїх роботах В.П. Юшин [5], де «система це ціленаправлена взаємодія елементів по заданій шкалі цінностей», рисунок 1. Виходячи з цього визначення система має ціль, елементи, взаємозв'язок цих елементів (комунікації) та шкалу цінностей за якою функціонує система та визначає рівень її раціональності.

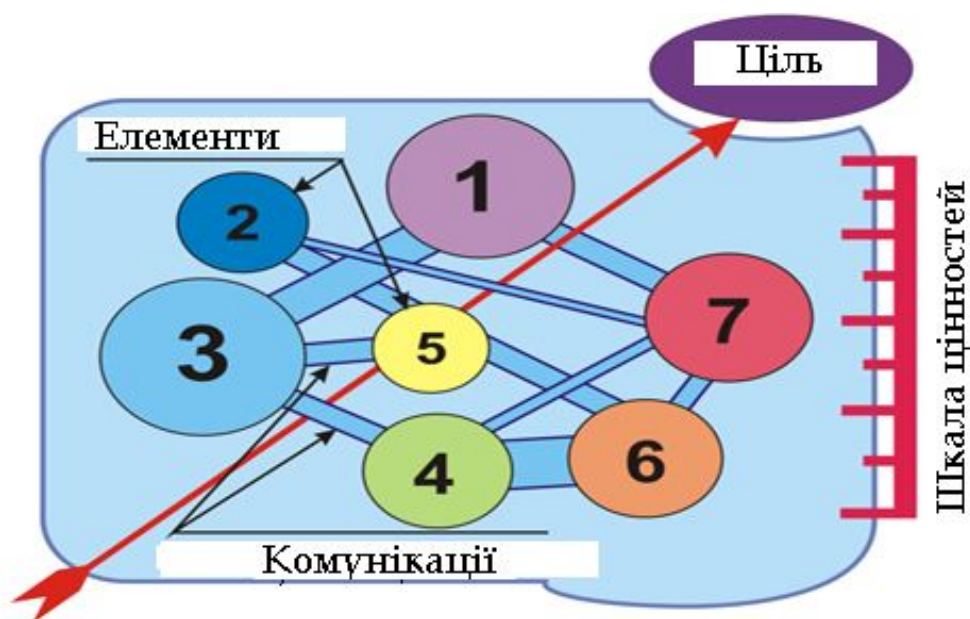


Рисунок 1. Система, як ціле направлена взаємодія елементів по заданій шкалі цінностей

Варто відмітити, що досить часто поняття система ототожнюють із поняттям структура. Це різні речі. Так як система відображає взаємодію елементів, тоді як структура ціленаправлений взаємозв'язок елементів. Система відображає певну дію, в той час як структура відображає будову, зріз, фотографію, або система в який фрагмент часу.

Для зміни системи необхідно змінити її ціль, елементи, рівень зв'язку (комунікабельність) та шкалу цінностей. Заходи направлені на заміну окремих елементів системи не приведуть до бажаного результату, а змінять тільки рівень функціонування системи не більше. Це стосується як костних, фрактальних так і соціальних систем, про які ми поговоримо в наступному підрозділі.

Класифікація систем та їх властивості

Формуючи класи систем, застосовують різні класифікаційні (системотворні) ознаки, головними з яких вважають природу та походження елементів, тривалість існування, мінливість властивостей (поводження), ступінь складності, відносини до середовища тощо. Одну з можливих класифікацій систем наведено в таблиці 1.

Таблиця 1

Класифікація систем

№ з/п	Класифікаційна ознака	Класи систем
1	Природа елементів	Реальні, фізичні, абстрактні
2	Походження елементів	Природні, штучні
3	Тривалість існування	Постійні, тимчасові
4	Мінливість властивостей	Статичні, динамічні, стохастичні, детерміновані
5	Ступінь складності	Прості, складні, великі
6	Ступінь стійкості	Стійкі (рівноважні), нерівноважні
7	Реакція на збурювальні впливи	Активні, пасивні
8	Характер поведіння	З управлінням, без управління
9	Ступінь участі в реалізації управлінських впливів людей	Технічні, людино-машинні (ергатичні), організаційні
10	Ступінь зв'язку із зовнішнім середовищем	Відкриті, закриті й ізольовані

Зупинимось докладніше на деяких основних типах систем.

Абстрактні системи складаються з елементів, що не мають фізичних аналогів у реальному світі. Наприклад, системи рівнянь, системи числення, ідеї, плани, гіпотези, теорії тощо.

Штучні – це системи, які створила людина.

Прості системи – такі, що їх можна описати з достатньою точністю.

Великі складні системи – складаються з численних взаємозалежних і таких, що взаємодіють між собою, різнорідних елементів та підсистем. Складні системи мають принципово нові властивості, яких не має жодний зі складових елементів (властивість емерджентності). Приклади складних систем: живий організм, підприємство, галузь економіки, система управління телекомунікаціями і т. ін. Такі системи характеризуються високим рівнем невизначеності свого поведіння.

Ізольовані (закриті) системи – на відміну від відкритих систем не обмінюються із зовнішнім середовищем енергією, речовиною або інформацією.

Організаційні системи – соціальні системи, групи, колективи людей, суспільство в цілому.

Кібернетичні системи – складні динамічні системи з управлінням. Кібернетична система – це множина взаємозалежних об'єктів (її елементів), здатних сприймати, запам'ятовувати і переробляти інформацію, а також обмінюватися нею. Приклади кібернетичних систем: автопілот, регулятор температури, комп'ютер, людський мозок, живий організм, підприємство, людське суспільство.

Кібернетичним системам притаманна низка нових властивостей, яких можуть не мати системи інших типів:

- 1) багатоваріантність поведіння;
- 2) керованість (інформаційним впливом на систему можна змінити її поведінка);

- 3) наявність пристрою керування;
- 4) здатність взаємодіяти з навколишнім середовищем як безпосередньо, так і через пристрій керування;
- 5) існування між системою, середовищем та пристроєм керування каналів інформації;
- 6) здатність інформації, яка циркулює по цих каналах, утворювати зворотні зв'язки, за допомогою яких здійснюється управління поведінкою системи з боку органів управління;
- 7) цілеспрямованість управління системою: воно спрямовує систему до вибору певного поведінки або стану, компенсуючи зовнішні збурення;
- 8) досягнення мети, так само як і поведінки системи, має ймовірнісний характер і визначається співвідношенням потужності збурювальних впливів та ефективності пристрою керування (здатність до переробки інформації та вироблення оптимальних у певному сенсі керувань впливів);
- 9) властивість рівноваги, притаманна деяким кібернетичним системам, тобто здатність пристрою керування повертати систему до початкового стану або до початкового поведінки, компенсуючи збурювальні впливи;
- 10) властивість самоорганізації, також притаманна деяким кібернетичним системам, тобто здатність відновлювати або змінювати свою структуру та спосіб функціонування, компенсуючи збурювальні впливи [6].

Таким чином, у підході, щодо класифікації та властивостей систем не менше літературного різноманіття ніж у визначенні самого поняття система.

Така класифікація вносить більше незрозумілості ніж хоча б якоїсь структурованості.

Класифікація систем проводиться в залежності від того яку мету перед собою ставить спостерігач чи дослідник.

Найбільш доцільною на нашу думку та найбільш універсальною класифікація повинна бути простою яка заключається в наступному: при

розгляданні різних процесів в природі їх умовно можна розбити на три великі групи: костні (мертві), живі та соціальні. Виходячи з цього ми виділяємо відповідно три найбільших типи систем: костні (мертві), живі та соціальні, всі інші типи можна віднести до цих трьох.

З нашої точки зору найбільш прикладними є наступні властивості системи: синергетика та емергентність.

Перед розкриттям цих понять нам необхідно зрозуміти поняття стану системи. Розрізняють два протилежних стани: врівноваженості та невраїноваженості. Врівноважені та невраїноважені стани можуть бути визначені наступним чином: розглянемо зміну функції Y при зміні аргументу X якщо малій зміні аргументу відповідає мала зміна функції, то маємо справу з станом врівноваженості, і навпаки, стан невраїноваженості зв'язаний з великою зміною функції при малому зміні аргументу [7]

В своїй роботі «От синергетики к информационной медицине» Дульнев Генадій Миколайович – професор, доктор технічних наук, дійсний член РАЕН, проводячи аналіз основної властивості системи синергетика виділяє три основні групи: костні (мертві), живі та соціальні системи [7]. Основна думка автора в даній роботі, що стосується властивостей систем заключається в наступному: в природі відсутні ізольовані системи, всі системи відкриті, тобто обмінюються з навколишнім середовищем масою та енергією, а відтак «... в ході розвитку невраїноважених процесів при деякому критичному значенні зовнішнього потоку енергії або речовини із-за втрати стійкості із невраїнованих можуть виникнути ввраїновані стани. Останні можуть носити просторовий, часовий або просторово-часовий характер. В цьому і заключається процес самоорганізації (синергетики), котрий відбувається як в костному, живому так і соціальному світі».

З нашої точки зору, другою найважливішою властивістю системи яка заслуговує уваги на розгляд це емергентність.

Емергентність - незведеність окремих властивостей системи котрі проявляє система в цілому до властивостей окремих її елементів. Це означає, у системи в цілому можуть проявитися такі властивості котрі не властиві жодному елементу системи окремо [7].

Властивість системи емергентність є тотожністю когерентності. Правда когерентність більш вживана при розгляді процесів пов'язаних із потоками світла та хвиль. Із поняттям когерентності як елементу синергетики у своїх роботах розкривають Винер И, [9]; Девятков Н.Д., Белицкий О.В.[10]; Крашенюк А.И., Крашенюк С.В.[11; Курис И.В. [12].

Розглянувши різноманітні тлумачення поняття система та їх властивостей підсумовуючи даний розділ приведемо ще одне визначення поняття системи, яке дає П.К. Анохин в розрізі її властивостей «Системою можна назвати такий комплекс вибірково-залучених компонентів, у яких взаємодія приймає характер взаємосприйняття («взаимоСОдействие») компонентів на отримання корисного результату» [8]. Для нашої системи за мету створення якої ми ставимо в даній роботі, таким корисним результатом буде високоефективне сприйняття даних від локальних цифрових медичних приладів, їх перетворення та передачу у web - інтерфейс.

1.2 Поняття «медприладів»

Головною і кінцевою метою діяльності всіх працівників охорони здоров'я є безпосереднє використання всіх доступних заходів для підтримування здоров'я пацієнта в оптимальному стані. Практичні лікарі добиваються саме цього.

Медичний прилад (далі медприлад) – це технічний пристрій, що призначений для спостереження, вимірювання, обробки і представлення інформації про живий організм та для передачі цієї інформації на відстань.

Медичний апарат – це технічний пристрій, що дозволяє генерувати різні енергетичні фактори, що діють на живий організм з лікувальною або терапевтичною метою та підтримувати нормальне функціонування організму.

Взагалі, поняття “ апарат “ походить від грецького слова apparatus (устаткування, прилад, технічний пристрій). Ми ж розділяємо це поняття на два: апарат і прилад. Спільний термін для медичних приладів і апаратів є медична апаратура.

Структурна схема медичного апарату медико-біологічної інформації.

Основним призначенням медичної апаратури є вимірювання фізіологічних параметрів, які, як правило змінюються з часом і які викликані фізіологічними процесами, що протікають в живих тканинах і органах. Вимірювання фізіологічних показників необхідно як при діагностиці захворювання так і при спостереженні за динамікою лікувального процесу. Його можна розділити на такі основні частини (рис.8.1):

- перетворювач,
- пристрій для обробки сигналу,
- пристрій для відображення інформації на дисплеї чи для запису на папері.
- джерело живлення.

Перетворювач – це частина приладу, що здатна сприймати вимірюючий фізіологічний показник і перетворювати його в електричний сигнал. Закон зміни цього електричного сигналу відповідає закону зміни вимірюючого показника. Наприклад, вимірюється артеріальний тиск, який викликає зміни у перетворювачі і приводять до появи напруги. Напруга збільшується із підвищенням тиску і зменшується із зменшенням тиску. В цьому випадку перетворювач перетворює неелектричну величину в електричну (напругу). Перетворювач потрібний і тоді, коли фізіологічний показник має електричну природу. Наприклад, зняття електрокардіограми чи електроенцефалограми

вимагає при вимірюванні перетворювача (електродів), які перетворюють іонну напругу в електричний сигнал.

Пристрій для обробки сигналу. В більшості випадків електричний сигнал, який поступає від перетворювача, повинен пройти обробку, перш ніж він прийме зручну форму для нормального його використання в пристрої відображення. Таку модифікацію здійснює пристрій обробки сигналів. Спочатку сигнал підсилюється за допомогою одного чи каскаду підсилювачів. Надалі електричний сигнал обробляється різними способами, якщо це потрібно для пристрою відображення.

Пристроєм відображення (дисплей чи самописець). Результати вимірювання повинні бути представлені в такій формі щоб лікар міг сприймати і зберігати їх. Це здійснюється за допомогою дисплея, самописця, осцилоскопа чи іншого пристрою або за допомогою різних їх комбінацій [13].

1.3. Поняття «трансформація даних»

Комплекс методів і алгоритмів, спрямованих на оптимізацію уявлення та форматів даних з точки зору вирішуваних завдань і цілей аналізу. Трансформація не ставить метою змінити інформаційний зміст даних. Її завдання - представити інформацію в такому вигляді, щоб вона могла бути використана найефективніше.

Взагалі, трансформація даних - це дуже широке поняття, яке не має чітко окреслених меж. В різних напрямках обробки даних цей термін іноді поширюють на будь-які маніпуляції з даними незалежно від їхніх цілей і методів. Проте в контексті аналітичних технологій трансформація даних має цілком конкретні цілі і завдання, а також використовує досить стабільний набір методів. До основних з них відносяться нормалізація, перетворення типів і форматів, сортування, угруповання, злиття і ін.

На кожному етапі того чи іншого процесу є свої пріоритетні цілі трансформації, наприклад, в системах оперативної обробки (OLTP) це забезпечення підтримки коректності форматів і типів даних, оптимізація процесів доступу і вивантаження даних. На етапі ETL-процесу трансформація виробляється з метою приведення даних у відповідність з моделлю, яка використовується в сховище, а також забезпечення процесу консолідації даних та їх завантаження в сховище. І нарешті, в аналітичному додатку виробляється безпосередня підготовка даних до аналізу, об'єднання і виділення найбільш цінної інформації, забезпечення коректної роботи аналітичних алгоритмів, методів і моделей.

Трансформація даних є одним з найважливіших процесів в аналізі. Від того, наскільки грамотно вона буде проведена, залежить ефективність аналізу, достовірність і точність його результатів, а в подальшому і правильно проведений синтез. [14].

Трансформація – перетворення даних з однієї форми на іншу для забезпечення погодженості їхніх елементів, розташовуваних у різних інформаційних джерелах. Трансформація може мати різні рівні складності: *базові трансформації*, наприклад, перетворення типів даних, маніпуляції з рядками й *прості обчислення*; перетворення середньої складності, наприклад, операції пошуку й заміни, агрегація, резюмування та ін.; *комплексні перетворення*: типу складного граматичного розбору. Крім того повинні надаватися засоби для розроблення додаткових або розширення наявних способів трансформації даних. [15].

1.4 Визначення поняття «База даних» (БД)

Корпорація EMC оприлюднила результати проведеного дослідження «Цифровой Вселенной» - Extracting Value from Chaos (Як отримати користь від

хаосу), що проводилось агентством IDC. Згідно звіту, об'єм інформації в усьому світі збільшується більш ніж в два рази кожні два роки. Дослідники прийшли до висновку, що чисельність ІТ-персоналу не відповідає швидкому зростанню серверів, потреб в управлінні даними та об'ємів файлів. У наступні десять років ІТ-департаменти по всьому світу можуть зіткнутися з безліччю проблем: серверів стане в 10 разів більше, об'єм інформації, якою потрібно керувати, виросте в 50 разів, в 75 разів зросте число файлів або контейнерів, в яких інформація інкапсулюється в "Цифровому Всесвіті", а чисельність ІТ-персоналу, який управляє всією цією інформацією і серверами, виросте всього в 1,5 рази [5].

Розглянемо основні поняттями та визначеннями, що використовуються в контексті даної роботи.

База даних – це впорядкований набір логічно взаємопов'язаних даних, які спільно використовуються та призначені для забезпечення інформаційних потреб користувачів.

Головним завданням БД є гарантоване збереження значних об'ємів даних та надання доступу до неї користувачу, або ж прикладній програмі.

БД складається з двох частин (завдань): збереженої інформації та системи управління нею.

До БД пред'являються наступні вимоги:

- інтегрованість даних, тобто зберігання даних у єдиному сховищі;
- незалежність даних. Розрізняють логічну (зміна схеми БД не вимагає коректування прикладних програм) і фізичну (зміна методу організації даних не впливає ні на прикладні програми, ні на схему БД) незалежність даних;
- адекватність БД предметній області, тобто можливість відображення будь-яких фактів, що характеризують предметну область, повнота й несутеречливість даних, актуальність інформації (відповідність її стану відображуваної реальної системи на даний момент часу);
- цілісність (задоволення пропонованим логічним вимогам) даних;

- можливість взаємодії користувачів різних категорій та у різних режимах, забезпечення високої ефективності доступу для різних додатків;
- мінімальна надмірність (дублювання) збережених даних, що забезпечує необхідну продуктивність БД;
- здатність БД до розширення;
- можливість пошуку по декількох ключах;
- «приятність» інтерфейсів і малий час на освоєння системи, особливо для кінцевих користувачів;
- забезпечення захисту даних від несанкціонованого доступу або випадкового знищення збережених даних. Припускає введення заходів щодо ідентифікації користувачів і контролю їх дій з погляду наданих повноважень, а також відновлення БД при апаратних збоях обчислювального комплексу;
- прийнятні економічні та фізичні характеристики функціонування БД (вартість обробки, час реакції системи на запити, необхідні машинні ресурси та ін.).

Системи управління базами даних (СУБД) - це комплекс програмних засобів, що призначені для створення структури нової бази, наповнення її вмістом, його редагування і візуалізації інформації.

Під візуалізацією інформації потрібно розуміти вибір відображуваних даних відповідно до заданого критерію, їх впорядкування, оформлення і подальша передача на пристрої виводу інформації, або передачі її каналами зв'язку [2;9].

СУБД гарантує несуперечність, цілісність, таємність і мінімальну надмірність даних у БД. Ефективність СУБД визначається швидкістю доступу до даних, раціональним використанням пам'яті обчислювального комплексу, простотою розробки прикладних програм, що призначені для доступу до даних з БД. У сукупності БД і СУБД, що взаємодіють між собою, утворюють *банк даних* [1;3].

1.5. Класифікація СУБД за типом даних керування

Часто зустрічається характеристика БД на основі певних параметрів або необхідних вимог, наприклад:

- значна кількість даних;
- незалежність даних;
- відкритий доступ до даних;
- підтримка транзакцій з гарантією відповідних властивостей;
- гарантована відсутність збоїв;
- одночасна робота з багатьма користувачами.

До комерційних баз даних відносяться:

- DB2;
- Informix;
- Oracle;
- SQL Server.

До баз даних з відкритим кодом:

- MySQL;
- Firebird;
- PostgreSQL

DB2 – це система керування базами даних компанії IBM.

Informix – ім'я відомої родини систем керування базами даних (СУБД), та американської фірми, колишнього виробника цього програмного забезпечення (також Informix Software). Пізніше, у 2001 підприємство перейшло у власність IBM і з того часу продукт належить та далі розвивається у IBM.

СУБД Informix виділяється високою надійністю і швидкодією, вбудованими засобами відновлення після відмов, наявністю засобів реплікації даних і можливістю створення розподілених систем. Підтримуються майже всі

відомі серверні платформи: IBM AIX, GNU/Linux (RISC and i86), HP UX, SGI Irix, Solaris, Windows NT (NT, 2000) та Mac OS.

Oracle Database – це об'єктно-реляційна система керування базами даних від Oracle Corporation.

Microsoft SQL Server – це комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів – Transact-SQL, яка була створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO відносно структурованої мови запитів (SQL) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

MySQL – це вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СУБД) з відкритим кодом була створена як альтернатива 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, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Порівняльна характеристика баз даних представлена в таблиці 2. Виходячи з міркувань зручності ми розділили їх на два великих типи: платні та безкоштовні

PostgreSQL – об'єктно-реляційна система керування базами даних (СУБД). Є альтернативою як комерційним СУБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СУБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно до інших проектів з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СУБД та впроваджувати у неї найновіші досягнення.

Таблиця 2

Порівняльна характеристика баз даних

Безкоштовні			Платні		
Назва	Плюси	Мінуси	Назва	Плюси	Мінуси
MySQL	Швидкість, гнучкість даних, довільна вибірка з БД, компактність виводу	Складність ви-користання деяких функцій через специфіку системи	ORACLE	Висока надійність, швидкість обробки даних, крос плат-форменість, велика кількість літератури, підтримка зі сто-рони розробників	Завищена ціна, важкий синтаксис, не підходить для малих і середніх проєктів
SQL Server 2008 R2	Просте віддалене управління, син-хронізація корпора-тивних даних, ефек-тивна розробка, простий процес проєктування	Працює тільки під Windows системи	---	---	---
Postgre SQL	Надійність при великих навантажен-нях, кросплат-форменість, швидко-дія, наслідування	Інсталяція, відсутність літератури рідною мовою, відсутність Intelliscense при програмуванні	---	---	---

Для створення бази даних Web ресурсів найкраще використовувати MySQL, її переваги наступні:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки;
- одна з найшвидкодійчих СУБД.

1.6 Система управління контентом WordPress

WordPress – це популярна система керування контентом сайту (CMS), яка має відкритий вихідний код та поширюється під ліцензією GNU GPL. На даний момент, WordPress найчастіше використовується для створення блогів на її основі. Також на основі даної CMS можна створювати достатньо складні новинні ресурси, а також інтернет-магазини. WordPress написана на мові програмування PHP та в якості бази даних використовує MySQL. Зараз доступно багато тем та плагінів, з допомогою яких можна створювати практично будь-які проекти. WordPress має досить просту панель адміністрації, яку легко освоїти за лічені години. А сам сайт, на основі WordPress, можна створити за один день. Серед можливостей WordPress можна виділити підтримку тем, плагінів, RSS, Atom, trackback, pingback, веб-стандартів (XHTML, CSS), SEO-оптимізації, наявність перекладів на різні мови, в т.ч. українську та інші можливості.

Переваги WordPress:

- стандартні з переваг, притаманні усім CMS, швидкий, мало займає місця, багато доробок та шаблонів, простий для зміни коду, гнучкий;

- найкраще для блогів (статей, думок, щоденників тощо);
 - легке налаштування віджетів (блоків), меню;
 - вбудований редактор сторінок HTML;
 - створення рекламних блоків та меню шляхом простого пересування їх мишкою у потрібне місце;
 - автоматичне додавання відповідних пунктів меню після створення нових публікацій у відповідному розділі;
 - гнучка та жива система; розробка не зупиняється;
 - вбудовані функції, які в інших CMS вимагають додаткових налаштувань, наприклад, коментування з дуже широкими можливостями;
 - вбудована і якісно реалізована можливість створювати зрозумілі посилання замість незрозумілих;
 - вбудована можливість відносити публікацію до декількох категорій (тем);
 - автоматичні оновлення усіх плагінів, тем та самої системи ВордПрес;
- Недоліки системи:
- зміна шаблону призводить до потреби заново будувати структуру сторінок за допомогою віджетів (блоки меню збоку);
 - відсутність єдиної системи побудови шаблонів; важко знайти підходящий шаблон (одночасно красивий та зручний);
 - бездумне використання таблиці wp_posts (та що містить статті);
 - з кожною зміною версії WordPress доводиться оновлювати плагіни;
 - відсутній нормальний форум для спілкування під WordPress.

1.7 Система управління контентом Joomla

CMS Joomla - популярна система управління контентом сайту, написана на мовах PHP і JavaScript, що використовує як сховища базу даних

MySQL. Є вільним програмним забезпеченням, поширюваним під ліцензією GNU GPL.

Інтерфейс панелі адміністратора Joomla інтуїтивно зрозумілий. Для того, щоб керувати Joomla, не потрібно бути «просунутим користувачем». Навіть «новачки» зможуть легко і швидко додавати і редагувати статті, новини, завантажувати графічні зображення й управляти іншим вмістом сайту.

Для CMS Joomla написано безліч безкоштовних розширень і модулів, що дозволяє легко розширити базові можливості цієї CMS і створювати серйозні інтернет-проекти.

На Joomla можливо реалізувати і простий сайт візитку, і багатофункціональний корпоративний сайт компанії, а так само великий Інтернет-портал або просто блог.

До першої і найвагомішої переваги Joomla слід віднести безкоштовність системи управління і безкоштовна підтримка сайтів на Joomla розробниками. Як і багато популярних CMS, Joomla поширюється безкоштовно. При цьому вона не поступається іншим, платним або умовно-безкоштовним CMS - наприклад, тому ж Бітрікс або DLE.

Не менш вагома наступна перевага - широкі можливості для створення сайтів.

Сучасне «сайтоформування» розвинене саме завдяки таким CMS - так як вони надають широкі можливості для розробників. За допомогою Joomla можна створити звичайний сайт, сайт-візитку або корпоративний сайт, блог - або інтернет-магазин. Для цього досить вибрати необхідні компоненти - і додати їх до системи управління.

Також одним з переваг (в основному це - підтримка сайтів) відносять невеликий об'єм займаної пам'яті вже встановленою системою - вона здійснює менше навантаження на сервер, бази даних (чим більшість інших безкоштовних систем) і, отже, сайт швидше працює.

Додатково до всіх своїх переваг, сайт на Joomla легко просувати і оптимізувати. Для зручної роботи та просування в пошукових системах також досить підібрати необхідні компоненти - або знайти фахівця, який буде цим займатися. До речі, підтримка сайтів на Joomla вважається однією з найбільш простих - так як легко знайти і грамотного оптимізатора, знайомого з даною CMS, і контент-менеджера.

Недоліки:

- невисока стійкість до навантаження;
- велика кількість готових шаблонів під Joomla спричиняє за собою появу сайтів "близнюків", що не завжди приємно, особливо якщо ви за цей сайт заплатили гроші;
- простота керування контентом є лише відносною, оскільки якщо Ви хочете реалізувати на сайті якісь нестандартні рішення у функціоналі або дизайні, то без досвідченого програміста не обійтися;
- іноді реалізація простого модуля спричиняє за собою безліч програмної роботи, яка в результаті може себе не виправдати;
- відкритий код і безкоштовність призводить до виникнення дір в системі безпеки сайту.

1.8 Система управління контентом Drupal

Drupal - це CMS, що означає система управління вмістом вашого сайту в Інтернеті. Дана система є повністю відкритою, з ліцензією GPL. Так як Drupal вільний у використанні, його написання може доповнити будь-який програміст. Система безкоштовна та багатофункціональна. Написана CMS на мові PHP і зберігає всі дані, використовуючи реляційну таблицю. Підтримує PostgreSQL, MySQL та інші види баз даних.

Технічними особливостями даної системи є:

- єдина таксономія сайту;
- кастомзовані типи матеріалу;
- модульну систему;
- розширену роботу з формами;
- великий набір стандартних функцій для оптимізації.
- пошук по вмісту сайту, в тому числі по таксономії і користувачам;
- різні властивості при побудові рубрикаторів;
- обмеження ролей;
- багатофункціональне створення меню;
- підтримка форматів XML;
- підтримка посилань зрозумілих для людей;
- можливість перекладу інформації сторінки і безпосередньо адміністративної частини на різні мови;
- можливість установки, як вбудованих шаблонів, так і професійно розроблених дизайнерських шаблонів;
- отримання повідомлень і легке оновлення нових версій системних модулів.

Drupal підтримують тисячі розробників і організацій зі всього світу. Він використовується практично у всіх формах онлайн розробок: від простих блогів до потужних комерційних, організаційних і управлінських об'єктів.

В Harvard, Rackspace, Yahoo, AT&T і навіть в Білому домі використовують Drupal. Інноваційна модульна архітектура ядра Drupal забезпечує його гнучкість, яка загальнодоступна далеко не у всіх інших веб-фреймворках чи CMS.

Коли ви впроваджуєте нову технологію у вашу бізнес модель, досить важливими є легкість її придбання, налаштування і можливість оновлення (бажано без доплати) при розширенні масштабів підприємства. Будь-які

труднощі з купівлею чи налаштуванням потребують додаткового часу чи грошей, або і часу і грошей. Кожен може зайти на сайт і закатати копію платформи Drupal і більшість розширювальних модулів безкоштовно. На сайт розміщено тисячі модулів, сотні тем і навіть інструменти управління сайтом з командної стрічки, для тих, хто хоче брати участь у розробці Drupal є тут. Загальнодоступні проекти Drupal щоденно додаються на сайт системи. Вони призначені для реалізації більшості корисних задач вашого сайту. Ви можете їх завантажити, розмістити в папці модулів і запустити декількома кліками мишки.

Говорячи про CMS Drupal не можна не назвати наступні її переваги.

Масштабованість - Drupal надає можливість створювати як прості сайти - візитки, так і серйозні проекти, серед яких соціальні мережі, інтернет-магазини, різного роду портали та новинні ресурси. Це не просто слова, а дійсно проаналізований теоретично (з літературних джерел) та підтверджений на практиці факт.

Гнучкість архітектури – одна з найвагоміших переваг, за яку варто подякувати всім тим, хто причетний до розробки системи управління Drupal. Можна тільки дивуватись та отримувати задоволення від витонченості роботи ядра даного «движка» і система хуків. У 99% випадків будь-яка зміна в архітектуру проекту можлива без написання власного модуля, або створення теми.

Універсальність і гнучкість - професійна CMS Drupal має дуже гнучке ядро і добре продуману архітектуру, що дозволяє легко створювати сайти різносторонньої направленості: інформаційні сайти, інтернет-магазини для різних товарів, корпоративні сайти, сайти знайомств, портали новин та соціальні мережі.

Ком'юніті - досить зайти на офіційний сайт Drupal, щоб зрозуміти наскільки потужна підтримка у даного продукту. Drupal.org - це цілий

«комбайн», де розміщуються всі основні модулі та загальнодоступні теми, є можливість створити Issue або Bug report, які не залишаються без уваги. Знову ж, з точки зору досвідченого розробника Drupal.org має чітку структуру і зберігає детальну інформацію щодо змін кожного проекту модуля або теми. Будь який модуль, викладений на офіційному сайті, спочатку проходить внутрішню систему тестів, а потім вручну модерацію. Ну і до того ж Drupal – соціальне об'єднання досить активне в плані проведення різного роду конференцій, інтернет обговорень і т.д.

Криптографія - зрозуміло, що це поняття суб'єктивне, адже все залежить від прямої рук розробника. Але, за нашим аналізом та власним досвідом, сайти на CMS Drupal терплять зломи набагато рідше в порівнянні з іншими системами управління контентом. Надійність і безпека - Drupal CMS забезпечує високу ступінь надійності. У ядрі системи застосовуються різні методи захисту котрі зводять ризик злому до мінімуму

Кешування - рано чи пізно доводиться стикатися з цим поняттям. Як правило, це відбувається на навантажених проектах. Що ж, Drupal і в цьому випадку забезпечив надійність роботи, надавши кілька зручних модулів і готових рішень для кешування. Кешування сторінок дуже допомагає для анонімних користувачів.

Швидкодія - Drupal застосовує різні методи для підвищення швидкодії сайту: кешування динамічних сторінок в статичні, кешування окремих повторюваних частин сайту, різні модулі для інтеграції з системами підвищення продуктивності, грамотна архітектура бази даних забезпечує швидке виконання запитів, оптимізація скриптів і стилів сайту для більш швидкого завантаження.

Порівняльна характеристика різних типів систем управління контентом сайту представлена в таблиці 3.

Таблиця 3

Порівняльна характеристика систем управління контентом сайту

Безкоштовні			Платні		
Назва	Плюси	Мінуси	Назва	Плюси	Мінуси
WordPress	Проста у використанні, зручне навігаційне меню, більше ніж 13000 безкоштовних модулів, взаємодія з пошуковими системами	Для отримання бажаного результату необхідно встановити велику кількість плагінів, часто оновлюється, що не зручно для адміністрування	Magento	Сама гнучка платформа для створення ESHOP, велика кількість функціоналу з «коробки», проста інтеграція з XML та CWS файлами	Низька швидкість мала кількість літератури, складність у програмуванні власних елементів, вузька спеціалізація
Joomla	Орієнтована на дизайнерів та розробників, велика кількість плагінів та інструментів, висока безпека системи	Труднощі з переходом на нову версію, часто піддаються атакам, складність в освоєнні, відсутність служби підтримки	---	---	---
Drupal	Гнучкість конфігурації, велика спільнота розробників, велика кількість модулів, система кешування	Довгий процес навчання, складний інтерфейс	---	---	---

Простота підтримки та обслуговування - велика кількість модулів і налаштувань дозволяє легко розширювати і допрацьовувати сайт. Найчастіше це можна зробити без програмування за допомогою панелі адміністрування. Якщо сайт на Drupal організований грамотно, то немає потреби "переробляти заново" або "мінати архітектуру" для продовження розвитку сайту. Висока гнучкість системи дозволяє реалізовувати вимоги швидше ніж інші системи, а отже дешевше.

Не можна не сказати про вільну роботу з власним кодом (створення своїх власних модулів) та відмітити прекрасну роботу типів матеріалів з власним кодом, а саме:

- вивід необхідної структури сайту із бази у необхідному нам вигляді;
- підвищення швидкодії сайту, завдяки зменшенню дерева структури;
- спрощення написання скриптів і т.д.

Серед недоліків системи управління контентом Drupal можна відмітити тільки два:

- складний інтерфейс;
- вимогливість до ресурсів веб сервера.

Ми відмітили два недоліки управління контентом Drupal, які, за певних умов можна оминати.

Коротка порівняльна характеристика найпоширеніших баз даних та систем управління контентом у вигляді таблиці знаходиться в додатку 1.

Висновки

- 1) Таким чином в даному розділі ми проаналізували різні визначення поняття система та зупинилися на наступному: система - ціленаправлена взаємодія елементів за заданою шкалою цінностей.

- 2) Проводячи класифікацію систем ми виділяємо відповідно три найбільших типи систем: костні (мертві), живі та соціальні, всі інші типи можна віднести до цих трьох.
- 3) З нашої точки зору найбільш прикладними є наступні властивості системи: синергетика та емергентність.
- 4) Синергетика – процес самоорганізації котрий відбувається як в костних, живих так і в соціальних системах. Суть синергетики в ході розвитку неврівноважених процесів при деякому критичному значенні зовнішнього потоку енергії або речовини із-за втрати стійкості із неупорядкованих можуть виникати впорядковані стани.
- 5) Емергентність - незведеність окремих властивостей системи котрі проявляє система в цілому до властивостей окремих її елементів.
- 6) Медичний прилад – це технічний пристрій, що призначений для спостереження, вимірювання, обробки і представлення інформації про живий організм та для передачі цієї інформації на відстань.
- 7) Що стосується системи керування реляційними базами при створенні універсальної системи трансформації даних медприлади – web інтерфейс з економічної точки зору, простоти програмування, підтримки та ефективності її роботи найбільш доцільно використовувати систему керування реляційними базами даних MySQL.
- 8) Зробивши аналіз найпоширеніших систем керування вмістом, ми вибираємо CMS Drupal, тому що, для розробки нашої системи дана CMS має необхідну достатність елементів та високий рівень комунікації між цими елементами.

Необхідна достатність елементів CMS Drupal включає в себе:

- кастомізовані типи матеріалу;
- таксонометричні словники;
- модульну систему;
- розширену роботу з формами;
- великій набір стандартних функцій для оптимізації.

- 9) Структура бази полів MySQL у CMS Drupal на наш погляд є досить зрозумілою та комфортною у використанні.

Список використаної літератури

- 1) <http://sek300i1.narod.ru/v/1.htm>
- 2) <http://ua-referat.com>
- 3) <http://leksika.com.ua/18941220/ure/sistema>
- 4) http://referatua.org.ua/category_content.php
- 5) Юшин В.П. Руководство по интенсивному развитию организаторских способностей руководителей. – Луцк: МП Зоря, 19980.- 256 с.
- 6) <http://buklib.net/books/22438/>
- 7) <http://resultsystem.com/platformofconcepts>
- 8) Анохин П.К. Очерки о физиологии функциональных систем, М.: Медицина, 1975.
- 9) Винер И., Кибернетика, М.: Советское радио – 1985.
- 10) Девятков Н.Д., Бецкий О.В. // Применение миллиметрового излучения низкой интенсивности в медицине.: ИРЭ АИСССР-1985 –с.6.
- 11) Крашенюк А.И., Крашенюк С.В., Материалы третьей конференции Ассоциации гирудологов (ред. к.м.н Птушкин В.В.) – 1993-с. 42-43.
- 12) Курис И.В., Танец в биоэнергопластике. СПб.: Балтийская Педагогическая Академия – 2004- с.112.
- 13) intranet.tdmu.edu.ua
- 14) <http://www.basegroup.ru/glossary/definitions/transformation/>
- 15) Шаховська Н.Б., Пасічник В.В. Сховища та простори даних. Монографія. –Львів: Видавництво Національного університету «Львівська політехніка», 2009.- 244 с.

РОЗДІЛ 2

СТРУКТУРА БАЗИ ТА ТИПИ ПОЛІВ СУБД MYSQL

2.1 Архітектура СУБД MySQL

Інформація, яка зберігається в базі даних, має вигляд таблиць з рядками і стовпцями. У кожному зі стовпців зберігаються дані різних типів, які задаються при створенні таблиці. У простих базах даних, як правило, можна сильно не замислюватися про тип збережених даних, але із зростанням бази це може стати великою проблемою. Від правильного опису типів даних може сильно залежати продуктивність сервера. Розглянемо на прикладі популярного веб - сервера баз даних MySQL, як за допомогою правильного вибору типів полів ми зможемо оптимізувати обробку даних в нашій базі даних.

Звичайно, в самому простому випадку будь-яку інформацію можна представити у вигляді символьного рядка, і таким чином зберігати всі дані, описавши так всі поля. Сервер MySQL підтримує автоматичний переклад даних в необхідний контекст - в залежності від операції, виробленої над полем. Наприклад, використовуючи арифметичний оператор "+", при складанні "3" + "4" ми отримаємо результат 7, як і очікували. Але чи варто говорити, що при цьому неявному перетворенні також потрібні машинні ресурси? Крім цього, не слід забувати також і про різницю в сортуванні символьних і числових даних. У першому випадку "9" буде в відсортованому списку стояти після "11", і щоб зробити сортування як для чисел, нам буде потрібно зробити перетворення даних поля при завданні умов сортування: `ORDER BY number + 0`. Крім витрат на неявне перетворення, слід також пам'ятати про те, що числові дані обробляються сервером баз даних завжди швидше, в порівнянні із символьними. Все це вказує на те, що нам слід задумуватися про обґрунтований вибір типів стовпців ще на етапі проектування бази даних. Варто використовувати типи, які займають менше місця. Дві головні задачі, які спонукають нас оптимізувати роботу сервера з базою даних - це швидкість і

об'єм займаного дискового простору. Дуже часто рішення цих проблем є взаємовиключними, але іноді навіть проста оптимізація даних може поліпшити обидві ці характеристики. Як уже згадувалося вище, чисельні дані завжди обробляються сервером баз даних швидше символічних, тому всі поля, в яких передбачається зберігати цифрову інформацію, слід створити з оголошенням відповідного типу: INT, FLOAT і так далі. Але і цього мало - треба відразу визначити діапазон можливих значень поля і задати тип, який буде займати якомога менше місця на диску. Це дозволить заощадити дисковий простір - адже на кожне значення у нас буде відводитися менше місця, - а також збільшить продуктивність, так як більш короткі поля будуть оброблятися швидше.

Для типу INTEGER існують синоніми з попередньо встановленими обмеженнями на розмір збережених даних: TINYINT, який займає 1 байт і дозволяє зберігати дані в діапазоні від -128 до 127; SMALLINT, розмір якого вже 2 байти, а діапазон даних від -32768 до 32768; MEDIUMINT, з розміром 3 байти; INT - 4 байти; BIGINT & mdash ; 8 байт. Для більшості невеликих проектів, навіть для каталогів продукції, цілком може вистачити діапазону чисел, що надається типом SMALLINT, тим більше що, оголосивши його беззнаковим (SMALLINT UNSIGNED), ми отримаємо діапазон можливих значень від 0 до 65535. Точно так само слід задуматися про використання типів для чисел з плаваючою комою, де тип FLOAT (4 байти) у використанні краще, ніж тип DOUBLE (8 байт). При проектуванні таблиці потрібно вибрати тип, який покриє всі потреби, але якщо на етапі експлуатації його діапазону не вистачить, то цю ситуацію завжди можна виправити за допомогою оператора ALTER TABLE, змінивши тип стовпця.

Типи з фіксованою довжиною обробляються швидше, але займають більше місця. Якщо у випадку цифрових даних зменшення розміру поля призводило як до збільшення продуктивності, так і економії дискового простору, то в разі символічних значень ситуація дещо інша. Для символічних полів можна використовувати типи CHAR для символічних полів з фіксованою

довжиною (рядок, яка займає менше місця, все одно буде займати все відведене для неї місце) і VARCHAR, TEXT (або BLOB, якщо при сортуванні даних необхідно враховувати регістр) для рядків з змінною довжиною, де один або декілька байтів виділено для зберігання довжини рядка. Рядки з фіксованою довжиною швидше обробляються сервером баз даних, а рядки із змінною довжиною займають менше дискового простору.

Однак при частій модифікації бази даних у випадку рядків зі змінною довжиною є небезпека фрагментації, що вимагає частішої оптимізації таблиці. Варто сказати, що для рядків з довжиною більше 255 символів ми можемо використовувати тільки типи з змінною довжиною - TEXT, MEDIUMTEXT і LONGTEXT (відповідно в них виділено 2, 3 і 4 байта на зберігання довжини рядка). Крім такого простого вибору між типами строкових даних, необхідно також ретельно проаналізувати збережену інформацію. Можливо, дані, що зберігаються в полі, можна представити у вигляді кінцевого списку з одним або декількома виборами значень. У цьому випадку можна скористатися типами ENUM і SET, які обробляються як цілочисельні типи, що дозволяє також збільшити продуктивність і заощадити дисковий простір.

Поля типу TEXT (BLOB) вимагають більш обережного поводження. Як правило, це поля, які містять великі обсяги інформації, і слід уникати як їх індексування, так і неявної їх вибірки за допомогою оператора SELECT *. Можливо, більш оптимальним рішенням при необхідності їх використання було б зберігати їх в окремій таблиці, витягуючи їх тільки при явній необхідності.

Для полів варто давати назву NOT NULL вказуючи, що вони не можуть мати порожнього значення. З одного боку, це економить 1 біт на рядок, а з іншого, дозволить збільшити продуктивність при виконанні запитів, так як сервер баз даних не буде перевіряти такі стовпці на предмет наявності порожніх значень. Більш того, в ранніх версіях MySQL стовпці, не оголошені, як NOT NULL, взагалі не могли індексуватися. Зараз ця ситуація виправлена, але цього вкрай не рекомендується робити. Необхідність використання значень

NULL зустрічається рідко, і завжди може бути обійдена завданням свого спеціального значення для формування задачі виходячи із ситуації, що поле не містить значення. Для цифрових даних це може бути 0, для символьних - порожній рядок "", для полів часу "0000-00-00" і так далі. У будь-якому випадку, це не набагато ускладнить обробку запитів, але дозволить виробляти її більш швидко.

Для зберігання часу і дат, варто використовувати спеціалізовані типи. Крім цифрових і символьних типів полів, в MySQL існують також типи полів для зберігання дати та часу: DATE, TIME, DATETIME, TIMESTAMP і YEAR. Обробка значень часу часто має свої особливості і логіку, тому буде не зовсім оптимальним створювати власний формат зберігання часу, і обробляти такі дані веб-скриптом. Варто пам'ятати, що набагато ефективніше передати обробку значень дати і часу самому серверу, тим більше що в MySQL є досить великий набір функцій для роботи з даними типу "дата" і "час" - як для обчислень, порівнянь і змін, так і для відображення та вилучення частини збереженої в полі інформації - наприклад, за допомогою функції YEAR () з поля DATETIME. Але при використанні цих типів теж варто задуматися, які з них або, можливо, їх комбінації вигідніше застосувати - поля DATE і TIME займають по 3 байти, в той час як поле DATETIME - 8 байт, але зберігання всієї тимчасової інформації в одному полі може бути більш прийнятним для подальшої обробки. Також не слід забувати про можливість використовувати поле YEAR, яке займає всього лише 1 байт.

2.2 Системи забезпечення зв'язку СУБД MySQL з програмними продуктами

СУБД MySQL - одна з безлічі баз даних, підтримуваних в PHP. Система MySQL розповсюджується безкоштовно і володіє достатньою потужністю для вирішення реальних завдань.

SQL - це аббревіатура від слів Structured Query Language, що означає структурована мова запитів. Ця мова є стандартним засобом для доступу до різних баз даних. Система MySQL являє собою сервер, до якого можуть підключатися користувачі віддалених комп'ютерів.

Для роботи з базами даних зручно користуватися засобом, що входить в комплект Web-розробника: Denwer phpMyAdmin, XAMPP. Тут можна створити нову базу даних, створити нову таблицю в обраній базі даних, заповнити таблицю даними, а також додавати, видаляти і редагувати дані.

В MySQL визначено три базових типи даних: числовий, дата і час, рядковий. Кожна з цих категорій поділяється на безліч типів. Основні з них:

INT	Ціле число
TINYINT	Маленьке ціле число (-127 до 128 або від 0 до 255)
FLOAT	Дійсне число з плаваючою точкою
DATE	Дата. Відображається у вигляді РРРР-ММ-ДД
TIME	Час. Відображається у вигляді ГГ: ММ: СС
DATETIME	Дата і час. Відображається у вигляді РРРР-ММ-ДДЧЧ: ММ: СС
YEAR [(2 4)]	Рік. Можна визначити дво-або чотирицифровий формат
CHAR (M)	Рядок фіксованої довжини М (М <= 255)
VARCHAR (M)	Рядок довільної довжини до М (М <= 255)
TEXT	Довгі текстові фрагменти (<= 65535)
BLOB	Великі двійкові об'єкти (зображення, звуки)

Кожен стовпець після свого типу даних містить і інші специфікатори:

NOT NULL - Всі рядки таблиці повинні мати значення в цьому атрибуті. Якщо не вказано, поле може бути порожнім (NULL).

AUTO_INCREMENT - Спеціальна можливість MySQL, яку можна задіяти в числових стовпцях. Якщо при вставці рядків у таблицю залишати таке поле порожнім, MySQL автоматично генерує унікальне значення

ідентифікатора. Це значення буде на одиницю більше максимального значення, вже існуючого у стовпці. У кожній таблиці може бути не більше одного такого поля. Стовпці з AUTO_INCREMENT повинні бути проіндексованими.

PRIMARY KEY - стовпець є первинним ключем для таблиці. Дані в цьому стовпці повинні бути унікальними. MySQL автоматично індексує цей стовпець.

Створення нової бази даних MySQL здійснюється за допомогою SQL-команди CREATE DATABASE.

```
CREATE DATABASE IF NOT EXISTS `base`
```

```
DEFAULT CHARACTER SET cp1251 COLLATE cp1251_bin
```

Створення нової таблиці здійснюється за допомогою SQL-команди CREATE TABLE. Наприклад, таблиця books для книжкового магазину буде містити п'ять полів: ISBN, автор, назва, ціна та кількість примірників:

```
CREATE TABLE books (ISBN CHAR(13) NOT NULL,  
PRIMARY KEY (ISBN),  
author VARCHAR(30),  
title VARCHAR(60),  
price FLOAT(4,2),  
quantity TINYINT UNSIGNED);
```

Щоб уникнути повідомлення про помилку, якщо таблиця вже є необхідно змінити перший рядок, додавши фразу "IF NOT EXISTS":

```
CREATE TABLE IF NOT EXISTS books ...
```

Додавання даних у цю таблицю здійснюється за допомогою SQL-команди INSERT. наприклад:

```
INSERT INTO books ( ISBN, author, title, price, quantity )  
VALUES ('5-8459-0184-7', 'Зандстра Мэт',  
«Освой самостоятельно PHP4 за 24 часа', '129', '5');
```

Для вилучення даних з таблиці служить оператор SELECT. Він витягає дані з бази, вибираючи рядки, які відповідають заданому критерію

пошуку. Оператор SELECT супроводжує чимала кількість опцій і варіантів використання.

Символ * означає, що необхідні всі поля. наприклад:

```
SELECT * FROM books;
```

2.3 Зв'язок PHP з MySQL

Попрацювавши з phpMyAdmin над створенням бази даних, можна приступити до підключення цієї бази даних до зовнішнього Web -інтерфейсу.

Щоб отримати доступ до бази даних з Web, використовуючи PHP, треба зробити наступні основні кроки : підключення до сервера MySQL, вибір бази даних, виконання запиту до бази даних, додавання, видалення, зміна, пошук, сортування, отримання результату запиту, від'єднання від бази даних.

Для підключення до сервера бази даних в PHP є функція `mysql_connect ()`. Її аргументи: ім'я комп'ютера, ім'я користувача та пароль. Ці аргументи можна опустити. За замовчуванням ім'я комп'ютера = `localhost` , тоді ім'я користувача і пароль не потрібно. Якщо PHP використовується в поєднанні з сервером Apache , то можна скористатися функцією `mysql_pconnect ()`. У цьому випадку з'єднання з сервером не втрачається після завершення роботи програми або виклику функції `mysql_close ()`. Функції `mysql_connect ()` і `mysql_pconnect ()` повертають ідентифікатор підключення, якщо все пройшло успішно. наприклад:

```
$link = mysql_pconnect ();  
if ( !$link ) die ("Невозможно подключение к MySQL");
```

Після того, як з'єднання з сервером MySQL встановлено, потрібно вибрати базу даних. Для цього використовується функція `mysql_select_db ()`. Її аргумент: ім'я бази даних. Функція повертає `true`, якщо зазначена база даних існує і доступ до неї можливий. наприклад:

```
$db = "sample";  
mysql_select_db ( $db ) or die ("Невозможно открыть $db");
```

Щоб додати, видалити, змінити та вибрати дані потрібно сконструювати і виконати запит SQL. Для цього в мові PHP існує функція `mysql_query ()`. Її аргумент: рядок із запитом. Функція повертає ідентифікатор запиту.

2.4 Система зв'язку з СУБД MySQL (phpMyAdmin)

PHPMyAdmin - веб -додаток з відкритим кодом, написаний на мові PHP і представляє собою веб - інтерфейс для адміністрування СУБД MySQL. PHPMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL і переглядати вміст таблиць і баз даних. Додаток користується великою популярністю у веб - розробників, оскільки дозволяє управляти СУБД MySQL без безпосереднього введення SQL команд, надаючи дружній інтерфейс.

На сьогоднішній день PHPMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, враховуючи всі нововведення СУБД MySQL. Переважна більшість українських провайдерів використовують цей додаток як панелі управління для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Додаток поширюється під ліцензією GNU General Public License і тому багато інші розробники інтегрують його у свої розробки, наприклад XAMPP, Denwer, AppServ, Open Server .

Проект на даному етапі його розвитку реалізований на більш ніж 62 мовах.

Можливості phpMyAdmin.

В даний момент phpMyAdmin дозволяє:

- створювати і видаляти бази даних;
- створювати, копіювати, видаляти, перейменовувати і змінювати таблиці;

- здійснювати супровід таблиць;
- видаляти , правити і додавати поля;
- виконувати SQL - запити , в тому числі пакетні SQL – запити;
- управляти ключами;
- завантажувати текстові файли в таблиці;
- створювати і переглядати дампи таблиць;
- експортувати дані у форматах CSV , XML , PDF , ISO / IEC 26300 - OpenDocument Text and Spreadsheet , Word , Excel і LATEX;
- адміністрування декількох серверів;
- управляти користувачами MySQL і привілеями;
- перевіряти цілісність посилальних даних в таблицях MyISAM;
- використовувати запит за зразком (Query - by - example - QBE), створювати комплексні запити , автоматично з'єднуючись із зазначеними таблицями
- створювати графічну схему бази даних у форматі PDF;
- здійснювати пошук в базі даних або в її розділах;
- модифікувати збережені дані в різні формати, що використовуються в встановлених функціях, наприклад, відображення BLOB - даних як зображень або як файли посилання і т.д.;
- підтримує InnoDB таблиці та зовнішні ключі;
- підтримує mysqli , покращене розширення MySQL.

Висновки

- 1) Таким чином, в MySQL визначено три базових типи даних: числовий, дата і час, рядковий.
- 2) Використовуваний нами популярний веб - сервер баз даних MySQL, за умови правильного вибору типів полів, забезпечує оптимізацію передачі даних в web-інтерфейс.
- 3) Числові дані обробляються сервером баз даних завжди швидше, в порівнянні із символьними, тому всі поля, в яких передбачається зберігати

цифрову інформацію, слід створити з оголошенням відповідного типу: INT, FLOAT і так далі.

4) Ще на етапі проектування баз даних здійснюється обґрунтований вибір типів стовпців. Варто використовувати типи, які займають менше місця. Дві головні задачі, вирішення яких забезпечить оптимізацію роботи сервера з базою даних - це швидкість і об'єм займаного дискового простору.

5) Визначений діапазон можливих значень поля і заданий їх тип забезпечують мінімальну зайнятість об'ємного дискового простору та збільшення продуктивності роботи MySQL.

Список використаної літератури

- 1) <http://progbook.ru/bd/mysql/>
- 2) По друпалу [19:52:31] Артем Гарлинский:
- 3) <http://www.ozon.ru/context/detail/id/5006821/>
- 4) <http://www.ozon.ru/context/detail/id/3817467/>
- 5) <http://www.ozon.ru/context/detail/id/19125291/>

РОЗДІЛ 3

ОПИС ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «ЗДОРОВ'Я»

3.1 Призначення, будова, технічні характеристики, принцип роботи програмно-апаратного комплексу «ЗДОРОВ'Я»

Багатофункціональний програмно-апаратний комплекс «Здоров'я» (далі апарат), призначений для здійснення діагностики, шляхом проведення комплексних вимірювань різними методами:

- вимірювання та аналіз процесів релаксації електрошкірного опору (ЕКС), акупунктурних зон людини для оцінки функціонального стану окремих систем і всього організму за способом В.В.Голубцова;
- акупунктурні діагностики за методом Накатані, Фоллю, чакрова діагностика;
- діагностика рівня соматичного здоров'я за методикою Г.Л. Опанасенко;
- тестові діагностики (іридодіагностика, східний анамнез, фізіономіка аюрведа, тестування препаратів, флори і алергенів) ;
- діагностика зору і слуху.

Апарат переносний і може бути використаний в амбулаторіях, санітарно - курортних , навчальних і науково-дослідних установах, лікарями сімейної медицини, а також у кабінетах лікарського контролю за особами які займаються фізичною культурою і спортом, реабілітацією здоров'я.

Передача даних здійснюється через USB порти комп'ютера (ноутбука).

Апарат розрахований на роботу при температурі від +10 до +35 С і відносній вологості до 80 %.

Конструктивно апарат складається з персонального комп'ютера, приставки, вимірювального щупа, периферійного обладнання та кабелів зв'язку

Загальний вигляд діагностичного пристрою представлено на фото 1.

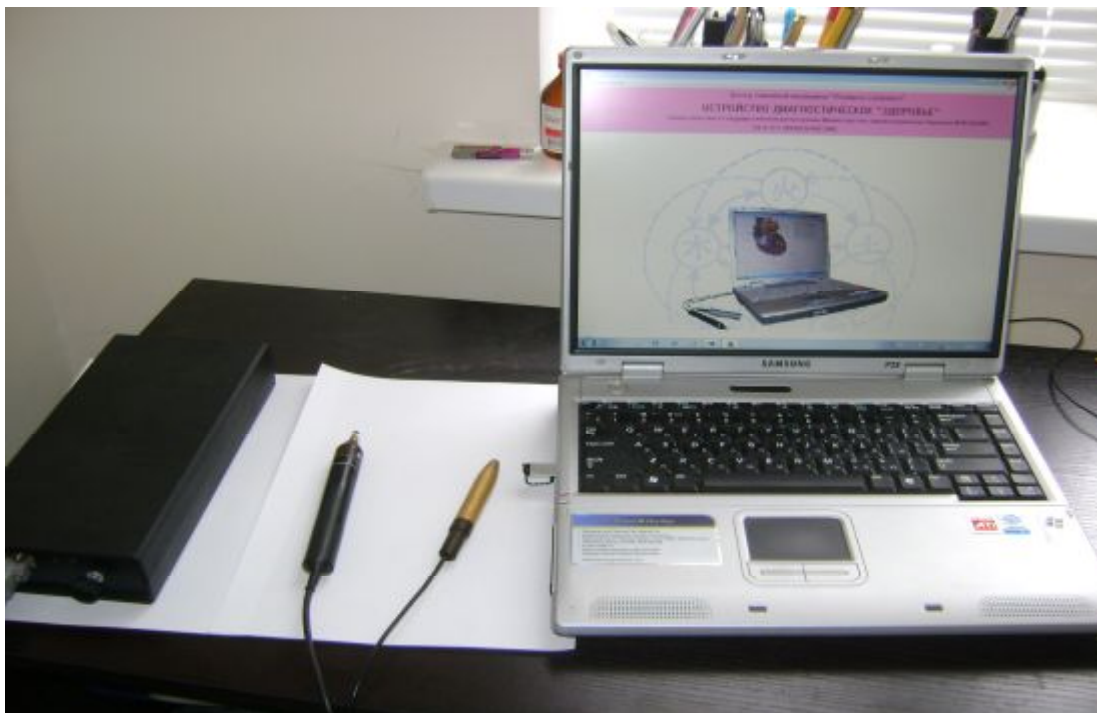


Фото 1. Загальний вигляд програмно-апаратного комплексу «ЗДОРОВ'Я»

Приставка складається з пластмасового корпусу та плати. Корпус складається з двох частин, виготовлених з ударостійкого полістиролу - нижньої і верхньої з'єднаних гвинтами. На бічних поверхнях корпусу розташовані кабель для підключення до USB портів, і кабель для щупів. Плата встановлена в середній частині корпусу з внутрішньої сторони.

Приставка 1 шт;

Комп'ютер (ноутбук) з приладдям 1 шт;

Принтер з приладдям 1 шт.;

Стандартний кабель USB;

Принтер 1 шт.;

Кабель електродів 9 pin - щуп з кнопкою і трубчастий загальний електрод 1 шт.;

Стереонавушники;

Диск інсталяційною програми 1 шт.;

Керівництво по експлуатації 1 шт.

Сумка для упаковки апарату 1 шт.

Вимірювальна напруга 1.5 вольт.

Напруга збудження 5.0 вольт.

Час збудження 0.6 сек.

Час релаксації 0.3 сек.

Повний час вимірювання однієї зони 1.0 сек.

Вимірювальний струм (ЕКС 20-1000 Ком) 1 - 50Ма.

Час встановлення робочого режиму після включення живлення не перевищує 1 хвилину.

Тривалість безперервної роботи апарату не менше 8 годин.

Тривалість циклу діагностичних вимірів не менше 0.5 години.

Споживання струму від комп'ютера не більше 100 мА.

Габарити апарату мм. не більше 100х190х40 .

Маса апарату без приладдя (кабелів і електродів) і пакувального футляра не більше 0.2 кг .

Робота апарату відбувається за допомогою комп'ютерного програмного керування по наступному циклу: первинний вимір постійного струму (ЕКС1) протікає між загальним електродом, електродом щупа і акупунктурною зоною людини під дією вимірювальної напруги 1.5 вольт, вплив на акупунктурну зону протягом 0.3 сек. напругою 5.0 вольт протилежної полярності і повторне вимірювання встановленого струму (ЕКС2) .

Послідовно накопичені дані корегуються відповідно до індивідуальних, статистичних (більше 10000 пацієнтів), вікових та інших особливостях пацієнтів. З двох значень ЕКС за допомогою спеціальних формул визначаються показники енергетичних і структурних порушень зон, з яких формуються показники для органів і систем.

За енергетичними характеристиками всіх зон проводиться додаткова побудова АУРИ енергетичної активності, де за допомогою методу динамічної модуляції параметрів визначаються енергетично застійні і активні органи і системи, які використовуються для корекції основних висновків програми.

Пристрій має два джерела живлення і дві землі. Мікросхеми мають

гальванічний зв'язок з комп'ютером живляться від стабілізатора 5 вольт, портів USB і підключені до машинної землі, вимірювальні ланцюги розв'язані за допомогою оптронів живляться з імпульсного стабілізатора +5 -5 вольт з власною землею.

Підготовка апарату до роботи.

Для отримання правильних результатів при вимірі і для попередження можливих пошкоджень апарату при користуванні ним слід дотримуватися наступних правил:

- до включення переконаватися в правильності підключення всіх кабелів зв'язку та заземлення
- включити шнур живлення до мережі
- запустити комп'ютер і дочекатися повного завантаження оболонки

Порядок роботи.

Провести подвійне клацання лівої кнопки миші по іконці тестової програми HEALTH. На екрані повинна з'явитися заставка програми з вікнами режиму вибору.

У вікні вибору слід клацанням лівої кнопки миші вибрати необхідний режим роботи (відкриття пацієнта з бази даних або реєстрація нового пацієнта). У разі відкриття пацієнта , наявного в базі даних можна зробити двома способами.

У вікні вибору за допомогою миші фіксується бажаний номер. Потім двічі клацнути лівою кнопкою миші по позиції реєстраційного файлу Reg.hdf. В результаті з'явиться інформаційне вікно з головним меню. Клацаючи лівою клавішею по позиціях головного меню активізується необхідний режим роботи.

В інформаційному вікні пацієнтів бази даних клацнути на відповідне прізвище.

У разі реєстрації нового пацієнта після натискання клавіші ENTER або клацання лівою кнопки миші по відповідній кнопці з'явиться допоміжне вікно для введення номера реєстрації пацієнта , куди слід вдрукувати бажаний номер і закрити вікно натисканням клавіші ENTER або клацнути лівої кнопки миші по позиції OK .

Після закриття вікна з'явиться система вікон для введення даних пацієнта. Перехід від одного вікна до іншого здійснюється кнопкою Tab , або клацанням лівої кнопки миші. Прізвище ім'я та по батькові друкуються повністю. Дата народження заповнюється у форматі 0x \ 0x \ xxxx . Стать вибирається клацанням лівої кнопки миші. Вікно закривається шляхом натискання кнопки ENTER.

Відключити шнур живлення комп'ютера від мережі . При цьому робота системи переходить в режим акумуляторного живлення.

Для проведення вимірювань акупунктурних діагностик необхідні два електроди, загальний і щуп .

Загальний електрод необхідно протерти спиртом і прикріпити до тіла в області пахової складки праворуч . У випадку дуже сухої шкіри загальний електрод можна обернути марлею змоченою в дезінфікованому водному розчині, прикріпити в цю ж зону щуп, попередньо протерти спиртом і обмотати чотиришаровою марлею, закріпити гумкою і звисаючи частини відрізати. Потім взяти в руку так , щоб кінчик вказівного пальця перебував на кнопці .

Кнопка знаходиться з правого боку щупа. Кінчик щупа слід змочити в губці, змоченою водним дезрозчином і щільно прикласти в необхідну точку.

Активуємо головне меню клацанням лівої кнопки миші, або натисканням клавіші F10 і вибираємо Діагностика \ акупунктура \ номер блоку точок . У результаті з'являється вікно зі схемним зображенням вибраної частини тіла з акупунктурними точками на ньому.

У правому нижньому кутку має бути інформаційне вікно з числом близьким до нуля. У лівому верхньому куті повинно бути порожньо. У разі наявності інформаційного вікна з підписом Data Present проведення вимірювань у даному блоці може носити тільки демонстраційний характер, так як дані вже існують і при закритті цього вікна виміряні дані будуть знищені.

Після встановлення в нижньому правому куті стабільного значення (ЕКС) необхідно натиснути на кнопку і відразу ж її відпустити не зрушуючи щуп з місця. При цьому будуть чутні два характерних мелодійних звуки з висотою

тону прямо пропорційному показникам електрошкірного опору. Щуп не знімати з точки до появи другого мелодійного звуку. Потім цикл вимірювання повторити для всіх точок блоку.

У процесі вимірювань колір точок на екрані змінюється з чорного на червоний.

Для повторного вимірювання якої-небудь точки необхідно натиснути клавішу ESC. При цьому колір точки зміниться і стане синім.

Після закінчення вимірювання точок сьомого блоку на екрані з'явиться два сімейства точок (графіки). Це візуальна інформація . Після закінчення перегляду, необхідно або натиснути Enter, або клацнути лівою кнопкою миші по кнопці ОК. Новий графік порівняльної характеристики енергетики окремих блоків і середньостатистичних показників необхідно також закрити звичайним способом. З'явиться зображення наступної групи точок у вигляді кривої, що відображає розподіл величини відгуку акупунктурних точок на їх збудження в заданому інтервалі показників, цю картинку також потрібно закрити натисканням на кнопку ОК. Після активізації головного меню звичайним способом необхідно вибрати Діагноз \ Енергетичне поле. Після натискання ENTER або клацання лівою кнопки миші по позиції меню на екрані з'явиться рядок для введення передбачуваного коефіцієнта посилення, де за замовчуванням стоїть 1.0.

Після натискання клавіші ОК на екрані з'явиться крива розподілу інтенсивності поля по зонах тіла пацієнта, причому внизу видно дві кнопки "Вихід" і "Введення". У разі, якщо обраний за замовчуванням коефіцієнт посилення не оптимальний необхідно натиснути на кнопку "Вихід" після чого на екрані з'явиться рядок для введення коефіцієнта посилення, куди необхідно за допомогою клавіатури занести нове (більше або менше) число.

Якщо коефіцієнт посилення вибраний невірно операцію отримання зображення слід повторити з іншим коефіцієнтом і потім натиснути кнопку " Введення". На екрані з'явиться малюнок з розподілом активності поля (вплив поля на зони органів пацієнта).

Після перегляду необхідно натиснути "Введення". Після закінчення аналізу на екрані з'явиться зображення розподілу активності поля в двох проекціях і три колонки систем організму з позначками ступеня впливу активності поля на них.

Закриття вікна АУРА здійснюється натисканням клавіші ENTER при цьому зображення записується у файл.

Потім необхідно активізувати головне меню (P10 або натискання лівої кнопки миші у верхній частині екрану) і послідовно розглянути графічну і текстову інформацію діагностичних висновків.

Друк текстової та графічної інформації здійснюється, або через меню Файл \ друк, або за допомогою редакторів WordPad і Paint шляхом вибору з бази даних текстових файлів Diagn.txt, DiagnD.txt і графічних файлів.

Графічне зображення даних параметрів здоров'я на папері може бути отримано шляхом створення зображення на екрані програмою і потім перенесенням її в буфер обміну комп'ютера (натискання клавіш ALT + PRINTSCREEN), подвійним клацанням по іконці програми Paint відбувається її запуск.

Необхідно увійти в головному меню в позицію "Вставити" і відзначити її натисканням лівої кнопки миші.

У відповідь з'явиться вікно яке запитує про зміну розмірів зображення, необхідно відзначити позицію "так". При цьому у вікні програми з'явиться потрібне зображення .

При виборі в меню друк з'являється проміжне вікно для встановлення параметрів друку. Необхідно увійти у "Властивості" і встановити альбомний друк після чого натиснути ОК .

Друк зображення функціонального стану внутрішніх органів проводиться, або з меню, або за допомогою редактора Paint шляхом вибору відповідних файлів типу BMP з бази даних BD та відповідного номера пацієнта.

Файли записуються в директорію номера пацієнта при перегляді програмою на екрані монітора. Після закінчення вимірювань необхідно зробити правильне закриття всіх вікон і вимкнути комп'ютер.

3.2 Аналіз результатів параметрів стану здоров'я людини

Аналогове значення електрошкірного опору акупунктурної точки програмним забезпеченням апарату перетворюється у цифрове, в даному випадку буквене.

Апарат робить самостійно приблизні висновки про стан основних органів та систем і можливих процесів що протікають в організмі, однак можна побудувати незалежні власні висновки використовуючи застосовану в програмі лікарську частину.

Апарат робить самостійно приблизні висновки про стан основних органів та систем і можливих процесів що протікають в організмі, однак можна побудувати незалежні власні висновки використовуючи застосовану в програмі лікарську частину. Лікарська частина представлена списком різних функцій організму із зазначенням їх стану на момент обстеження. Ця частина є основною для постановки діагнозу і несе в собі основну діагностичну інформацію. У ній же міститься розшифровка і конкретизація більшості позицій, що фігурують в інших частинах текстової та графічної інформації.

Навпроти списку функцій розташовуються дві колонки, в першій з яких міститься інформація про стан даної функції на момент обстеження, в другій-характеристика тяжкості патологічного процесу, якщо такий має місце і впливає на прояви даної функції, а так само стану анатомічного субстрату.

Позначення «Н» у першій колонках говорить про нормальну функцію; позначення «С» означає, що для даного елемента присутні початкові зміни функції за рахунок зниження адаптаційної активності зони; позначення «ф (+) >> - означає збудження;

позначення «ф (-) >> - про пригнічення. Кількість плюсів і мінусів, нестійке від 1-го до 3-х позначає ступінь зміни функції.

У другій колонці позначення «Н» говорить про нормальний стан анатомічного субстрату, в поєднанні зі зміною функції - про нещодавно виниклі процеси.

Позначення «П (1) >>», «П (2) >> і «П (3) >> вкажуть на різні стадії

порушення анатомічної будови органу. Позначення «С» у другій колонці означає, що для даного елемента присутні початкові паталогічні зміни за рахунок зниження адаптаційної активності точки.

Відзначимо, що утилітарний спосіб інтерпретації отриманих даних неприпустимий. Позиції оцінюються як нормальні або патологічні залежно від стану інших позицій.

Інтерпретувати отримані дані може тільки лікар, який пройшов курс навчання і має підходи сімейної медицини.

Висновки

1) Власне програмно-апаратний комплекс складається з 2-х частин:

- апаратного забезпечення;
- програмного забезпечення.

В свою чергу апаратне забезпечення включає в себе:

- приставка 1 шт: (аналоговий пристрій для прийняття показників електрошкірного опору акупунктурної точки);
- комп'ютер (ноутбук) з приладдям 1 шт;
- принтер з приладдям 1 шт.;
- стандартний кабель USB;
- кабель електродів;
- щуп з кнопкою і трубчастий загальний електрод 1 шт.

2) Принцип роботи давачів параметрів фізіологічного стану людини: залежність величини електрошкірного опору акупунктурної точки від впливу нормованого струму, результати якого обробляються програмним забезпеченням побудованим за методом Голубцова В.В.

3) Програмне забезпечення апарату розроблене на базі об'єктно орієнтованої мови програмування DELPHI. Воно включає в себе дві частини:

- графічна оболонка (інтерфейс лікаря);
- системи об'єктів (кнопок) на які написані обробники для виконання

тих чи інших дій: обробка та збереження даних в архів по візиту пацієнта у XML форматі без, на жаль, подальшої передачі даних у зовнішні системи.

4) На етапі доводки розроблюваної системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL передбачається розробка функціоналу по передачі архіву з XML файлами у зовнішні системи від програмного забезпечення апарату.

Список використаної літератури

1) Інструкція по роботі та експлуатації програмно-апаратного комплексу «ЗДОРОВ'Я»

РОЗДІЛ 4

АЛГОРИТМ РОБОТИ СИСТЕМИ ОТРИМАННЯ І ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ З ІНТЕРЕНТ ІНТЕРФЕЙСОМ НА СУБД MySQL

4.1 Огляд баз даних для створення алгоритму

При створенні універсальної системи передачі даних від мед-приладів у web-інтерфейс з економічної точки зору та простоти програмування, підтримки найбільш доцільно використовувати систему керування реляційними базами даних MySQL та модульну систему керування вмістом Drupal.

MySQL – це вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СУБД) з відкритим кодом була створена як альтернатива 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, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

PostgreSQL – об'єктно-реляційна система керування базами даних (СУБД). Є альтернативою як комерційним СУБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СУБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно до інших проектів з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СУБД та впроваджувати у неї найновіші досягнення.

Для створення бази даних Web ресурсів найкраще використовувати MySQL, її переваги наступні:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;

- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки;
- одна з найшвидших СУБД;

Drupal - це CMS, що означає система управління вмістом вашого сайту в Інтернеті. Дана система є повністю відкритою, з ліцензією GPL. Так як Drupal вільний у використанні, його написання може доповнити будь-який програміст. Система безкоштовна та багатофункціональна. Написана CMS на мові PHP і зберігає всі дані, використовуючи реляційну таблицю. Підтримує PostgreSQL, MySQL та інші види баз даних

З нашої точки зору, CMS Drupal є найбільш оптимальним варіантом, так як для розробки нашої системи дана CMS має необхідну достатність елементів:

- кастомізовані типи матеріалу;
- таксонометричні словники;
- модульну систему;
- розширену роботу з формами;
- великій набір стандартних функцій для оптимізації.

Структура бази полів MySQL у CMS Drupal на наш погляд є досить зрозумілою та комфортною у використанні [1]. Більш детально дане питання розглядалося у попередніх розділах.

При розробці алгоритму роботи створюваної системи ми забезпечили можливість реалізації основних етапів без яких не може бути побудована жодна система, а саме: підготовка; запуск; доводка; розвиток.

Етап підготовки. Перше питання, на яке ми повинні для себе дати відповідь на етапі підготовки: що система повинна робити? друге запитання: як система повинна це робити?

Наша мета: розробити систему обробки, збереження та відображення інформації (отримуваної від програмного забезпечення локального медичного пристрою у графічно-текстовому вигляді глобальної мережі Інтернет). Тобто наша система повинна виконувати обробку отриманих даних, після обробки – їх збереження на сервері в «облачному» сховищі з інтернет інтерфейсом, котрий

забезпечує можливість подальшого їх відображення.

Створюваний нами програмний продукт забезпечить виконання поставленої задачі, або відповідь на питання: як система буде працювати?:

- створюючи РНР аналізатори для обробки вхідної інформації;
- реалізуючи збереження даних у БД MySQL;
- забезпечуючи посилення запитів у БД та отримання відповіді;
- розробляючи функцію обробки отриманої відповіді та відображення її

у вигляді стовпчастих графіків.

Друге питання на яке ми повинні дати відповідь, що ми повинні зробити для того аби наша система забезпечувала високоефективне виконання саме цих функцій.

На етапі підготовки проводяться всі необхідні роботи для розробки даної системи, а саме: визначити поле діяльності даної системи, тобто сформувані перелік всіх робіт, чого не можна не зробити у відповідності із технологією отримання кінцевого продукту. Таким чином ми отримаємо перелік дій які забезпечать нам очікуваний результат. Перетворюючи даний перелік у чітку послідовність ми отримаємо алгоритм роботи нашого програмного продукту.

При розробці алгоритму створюваної системи ми керувались наступними критеріями функціонування системи:

рівновага: система повинна забезпечити врівноважувану роботу не залежно від кількості вхідних даних;

стійкість: стабільність роботи системи при зміні вхідних даних (параметрів); величина вихідного сигналу (параметру) за будь-яких обставин повинна прирівнятись до вхідного.

мінімальний час перехідного процесу: час за який система забезпечить повернення величина вихідного сигналу (параметру) до вхідного.

адаптація: при зміні вхідних даних, система в автоматичному режимі створює необхідну достатність нових елементів, пристосовується до нововведень, що забезпечує високу ефективність та якість її функціонування;

ієрархічність: кожен розроблений компонент може розглядатись як система (підсистема) ширшої глобальної системи;

Виходячи з вище вказаних завдань та критеріїв їх вирішення створений нами алгоритм роботи системи отримання і збереження інформації з інтернет –інтерфейсом на СУБД MySQL буде наступним:

- завантаження архіву з XML файлами від ПЗ апарату;
- отримання вмісту архіву;
- передача наданих із XML файлів у PHP;
- обробка XML файлів, перетворення їх вмісту у масив;
- створення черги;
- запуск батча;
- парсинг даних;
- трансформація даних з ПЗ апарату до вигляду у системі;
- збереження даних у необхідному вигляді використовуючи тип матеріалу (створення користувачів, “візитів”);
- генерація інтерфейсу: графіки візиту, графіки динаміки, рекомендації та мобільний Android додаток.

Більш детально дане питання розглядалось в розділі 1, тому ми зупинимось в даному розділі тільки на питанні етапах розробки алгоритму.

Розроблений нами алгоритм роботи системи отримання і збереження інформації з Інтернет – інтерфейсом на СУБД MySQL представлений на схемі 1.

4.2 Завантаження архіву від програмного забезпечення медприладів з XML файлами

Першим етапом в роботі комплексу є завантаження у веб-інтерфейс архіву з вхідними даними – файлами XML. Для завантаження у веб-інтерфейсі створена відповідна веб-форма. Загальний вигляд створеної веб-форми для завантаження архіву з XML файлами представлена на рисунку 1. Дана форма забезпечує високоефективне завантаження архіву на сервер. Алгоритм очікує завантаження архіву з XML файлами візиту пацієнта, через створену веб - форму.

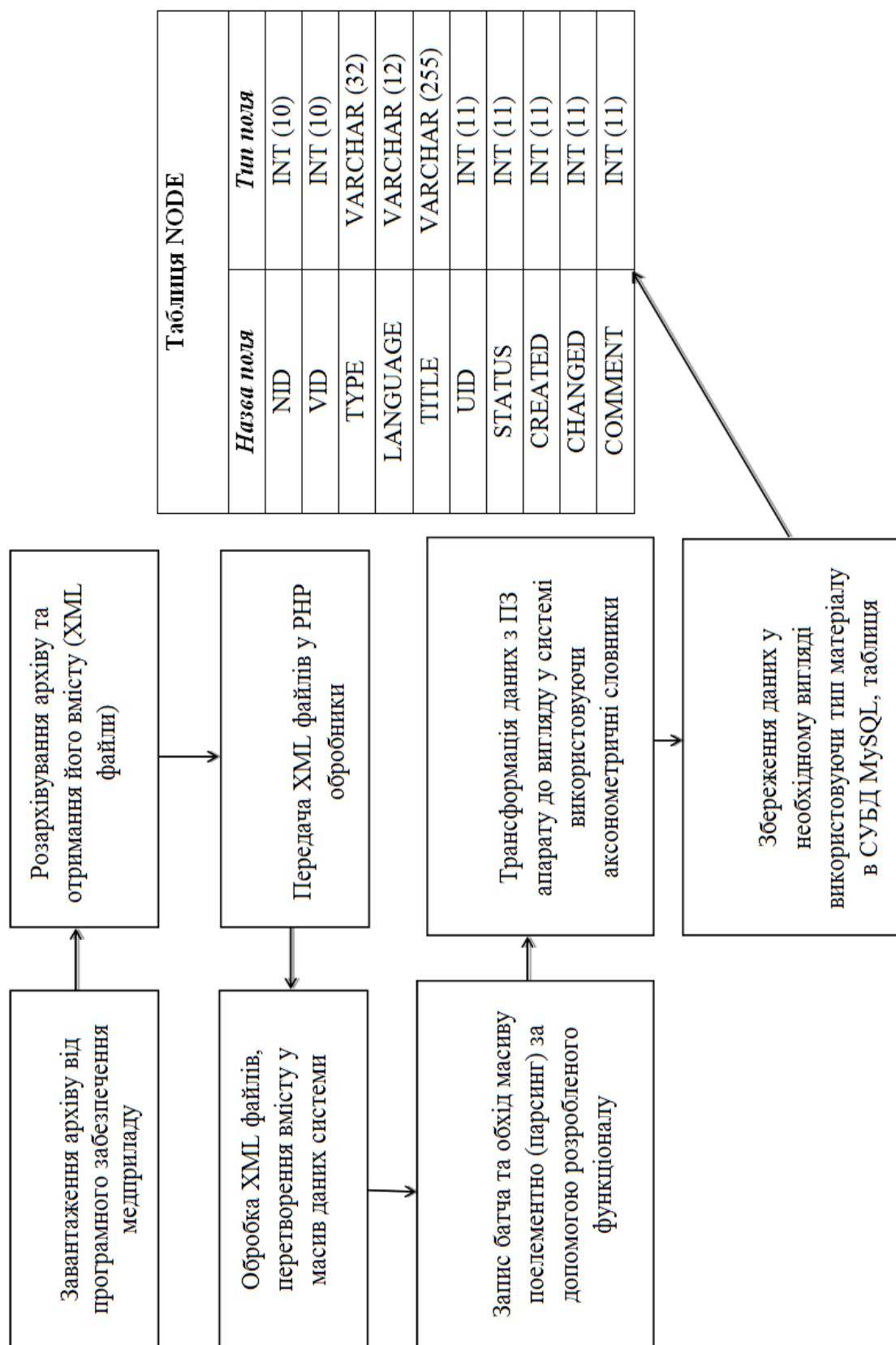


Схема 1. Алгоритм роботи універсальної системи трансформації даних: медприлади - web-інтерфейс.

Після активізації режиму “Відвантажити” програмний продукт зберігає файли на сервері. Тільки після того, як файли буде збережено запускається батч, котрий “обходить” інформацію у файлах обробляє її і записує у БД MySQL, та одразу створює ноду у системі з отриманих даних.

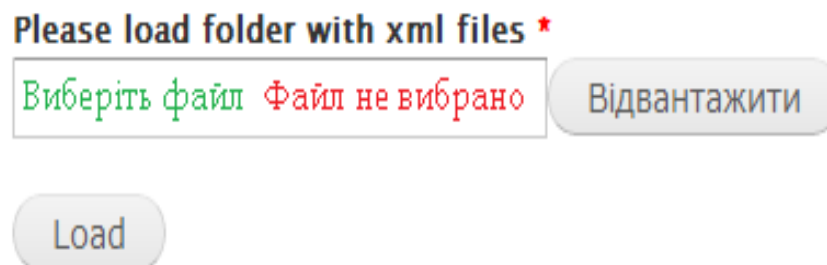


Рисунок 1. Веб - форма завантаження XML файлів з «візиту» пацієнта у web-інтерфейс.

Батч використовується для того, щоб розділити файл на менші частини тим самим зменшити навантаження на сервер. Після завантаження, архів із тимчасової папки переміщується в папку для подальшої обробки.

4.3 Розархівування архіву та отримання його вмісту (XML файли)

Для отримання вмісту архіву нам необхідно встановити на сервері декілька розширень.

Підтримка RAR прийшла в PHP через розширення PECL ext / rar, підтримуване Antony Dovgal і надає read-only API орієнтоване на функціях для лістингу та розпакування файлів RAR-архівів. Розширення LZF, підтримуване Marcin Gibula, дозволяє архівувати і розархівувати LZF, тоді як розширення Bzip2 дозволяє робити архівацію і разархівацію для Bzip2.

Найбільш простий спосіб установки цих пакетів це автоматизований встановлювач PECL, який дбає про скачування та встановлення розширень як завантажувального модуля PHP.

Ця процедура повинна створити завантажуваний модуль РНР з ім'ям `rar.so` у вашому каталозі розширень РНР. Тепер нам потрібно увімкнути розширення у файлі конфігурації `php.ini`, перезапустити веб-сервер і перевірити, чи доступне розширення, швидким викликом `phpinfo()`.

Приклад ідентифікації підтримування сервером відповідних розширень представлено в таблиці 1.

Таблиця 1

Приклад ідентифікації підтримування сервером відповідних розширень

Directive	Local Value	Master Value
<code>pgsql.allow_persistent</code>	On	On
<code>pgsql.auto_reset_persistent</code>	Off	Off
<code>pgsql.ignore_notice</code>	Off	Off
<code>pgsql.log_notice</code>	Off	Off
<code>pgsql.max_links</code>	Unlimited	Unlimited
<code>pgsql.max_persistent</code>	Unlimited	Unlimited

rar

Rar support	enabled
Revision	\$Revision: 1.18 \$

Reflection

Reflection	enabled
Version	\$Id: php_reflection.c,v 1.164.2.33.2.37 2007/04/12 18:39:46 johannes Exp \$

Щоб включити підтримку LZF, необхідно виконати наступне: скачати, скомпілювати і встановити розширення, використовуючи PECL або вручну. Для включення підтримки Bzip2 нам буде потрібно переструктурувати РНР з опцією `--with-bz2`.

Користувачі Windows можуть зробити це більш швидко. Попередньо скомпільовані версії файлів `php_rar.dll`, `php_lzf.dll` і `php_bz2.dll` можна завантажити на сайті <http://pecl4win.php.net/>. Як тільки ми отримали файли, розміщаємо їх в папку розширень РНР, активуємо їх через конфігураційний

файл `php.ini`, перезапустіть ваш веб-сервер. Тепер ви можете бачити активоване статус розширень в `phpinfo()`, як показано вище.

Припустимо, у вас є всі частини і вони працюють. Даваймо задіємо їх в простому прикладі: читання існуючого файлу RAR через PHP, вивід списку його змісту.

Після завантаження у веб-інтерфейс, архів з вхідними даними – файлами XML необхідно отримати вміст архіву – його розархівувати. Як тільки завантажений архів потрапляє на сервер він (архів) миттєво піддається процедурі розархівування. Дана процедура проходить з використанням спеціальних інструментів у чітко визначену папку. Після завершення процедури розархівування архів автоматично видаляється з метою не нагромадження інформації на сервері. Загальна процедура відкриття вмісту архіву:

```
<?php
```

```
// відкриття архиву
```

```
$rar = rar_open(realpath('.') . "/app-0.2.rar") or die('ERROR: Could not open archive!');
```

```
// отримання списку файли в у архіві
```

```
$entries = rar_list($rar);
```

```
foreach ($entries as $entry) {
```

```
    printf("%s (%d bytes)", $entry->getName(), $entry->getUnpackedSize());
```

```
    print "\n";  
}
```

```
// закриття архиву
```

```
rar_close($rar); ?>
```

Цей скрипт починається з використання функції `rar_open ()` для відкриття файлу RAR архіву. Якщо воно сталося, цей метод повертає хендл ресурсу, який представляє RAR архів. Цей хендл служить вхідний точкою для всіх RAR функцій. Потім використовується метод `rar_list ()` для видачі масиву, який зберігає інформацію про вміст архіву. Кожен елемент цього масиву представляє окремий файл (або елемент) з архіву, який зберігається як об'єкт. Методи об'єкта, такі як `getName ()` і `getUnpackedSize ()` використовуються для отримання додаткової інформації по відповідному файлу. Як тільки весь масив оброблений, хендл ресурсу знищується викликом методу `rar_close ()`.

4.4 Передача даних із XML файлів у PHP

XML (Extensible Markup Language) - це розширювана мова розмітки, що є підмножиною мови SGML і тому має спільну з ним мету - розмітка будь-якого типу документів.

XML має багато спільного з мовою розмітки HTML, проте вони різні за своїм призначенням. HTML, як і XML, є підмножиною SGML, але призначений тільки для відображення інформації в браузері. XML же призначений для структурованого зберігання інформації і не містить жодних конструкцій для відображення цих даних.

Для відображення даних, що зберігаються в XML-форматі, використовуються різного роду перетворювачі. Комбінація XML + перетворювач дозволяє досягти того ж результату, що і використання HTML-формоутворюваного відображення документа користувачеві. Однак у разі використання XML і перетворювача дані зберігаються окремо.

XML дуже схожий на HTML зі свого синтаксису, за винятком того, що теги XML можна придумувати власні. За аналогією з HTML всередині XML-тегів можна використовувати атрибути. Кожен тег XML повинен мати відповідний закриваючий тег (в HTML це не є обов'язковим)[2].

Для роботи з XML-документами ми використовуємо мову PHP. В PHP

для цього існує два модуля, реалізують два різних стандарти обробки XML - даних: SAX (Simple API for XML) і DOM (Document Object Model).

Стандарт SAX не є стандартом W3C і описує метод обробки XML - документів з метою отримання даних з них [3]. Тобто цей метод обробки XML - документів дозволить тільки прочитати дані з XML-документа, і не більше того. Створювати і змінювати XML-документи з його допомогою неможливо.

SAX заснований на так званому подієвому програмуванні. Його особливість полягає в тому, що ви надаєте парсеру XML набір власних функцій, які займатимуться обробкою різних типів XML даних (елементів (тегів), тексту і т.п.), а парсер потім буде сам викликати ваші функції в процесі обробки XML-документа, передаючи їм знайдені дані. Функції будуть викликатися в тій же послідовності, в якій відповідні дані розташовуються в XML-документи.

Іншим стандартом для обробки XML -даних є DOM – стандарт W3C, специфікацію якого можна знайти на сайті консорціуму <http://www.w3c.org/DOM>. На відміну від SAX, цей метод дозволяє виробляти будь-які операції з XML-даних з досить зручною формою – представляючи XML -документ як дерево об'єктів.

Модуль, який реалізує цей стандарт, називається DOM XML. Він не входить в основний набір модулів PHP, але може бути встановлений як розширення. API цього модуля намагається якомога більш точно слідувати стандарту DOM level 2. Крім того, існує безліч додаткових функцій. Ці функції включені для сумісності з попередніми версіями розширення, і використовувати їх у нових скриптах не рекомендується. Крім того, у розширення DOM XML є проблеми з російською кодуванням. Парсер обробляє текст тільки в кодуванні UTF-8, тому текст потрібно щоразу перекодувати за допомогою функції `iconv`. Звідси і необхідність встановлення розширення `iconv` разом з DOM XML. Модуль DOM XML є потужним і зручним у використанні засобом обробки XML - документів. В своїй роботі ми будемо використовувати саме цей стандарт обробки XML –документів.

Взаємодія PHP і XML відбувається за допомогою DOM XML.

Що відбувається, якщо взаємодія PHP і XML здійснюється за допомогою об'єктної моделі стандарту DOM? Модуль DOM XML визначає в PHP кілька класів, таких як DomNode, DomDocument, DomElement, DomText і DomAttribute, більшість з яких йдуть з ядра стандарту DOM. Майже для всіх класів (зокрема, для перерахованих вище) клас DomNode є батьківським, тому його властивості та методи успадковуються всіма іншими класами.

Якщо розглянути довільний XML-документ, то класу DomDocument відповідатиме сам цей документ, класу DomElement - кожен XML-тег, класу DomAttribute - атрибути тегів, а класу DomText - вміст XML-елементів. Водночас класу DomNode відповідатиме кожен з перерахованих елементів XML-документа.

Основним завданням даного етапу розробки алгоритму є формування масиву для його обробки. В результаті на виході отримуємо для кожного файлу свій масив даних, з якого формується «черга» обробки даних.

Для побудови черги нами розроблена унікальна функція «батч», яка суттєво зменшує навантаження на сервер та забезпечує швидкодію веб-інтерфейсу під час обробки отриманих від ПЗ апарату даних.

4.5 Обробка XML файлів, перетворення вмісту у масив даних системи

Останнім часом стало модним говорити про XML - розширювана мова розмітки (eXtensible Markup Language). Прийнято вважати, що він зможе вирішити багато проблем не тільки Web, але й інших інформаційних технологій, наприклад мов інформаційного пошуку.

Насправді перехід на стек специфікацій XML дозволяє:

- окремо описувати структуру даних без прив'язки їх до форми відображення,

- визначати форму представлення даних незалежно від конкретного змісту,
- створювати метадані,
- управляти доступом до даних,
- фільтрувати дані за змістом структури і метаданих,
- перетворювати зміст і структуру даних,
- погоджувати практику застосування даних з національним законодавством,
- описувати стандартні комунікативні формати даних, прийняті в різних предметних областях,
- та багато іншого.

Таким чином буде реалізовуватися даний етап алгоритму роботи розроблюваної нами системи.

4.6 Запис батча та обхід масивна поелементно (парсинг) за допомогою розробленого функціоналу

Багато хто знає про дуже потужний Batch API Drupal'a. Пакетні операції дозволяють виробляти маніпуляцію відразу з великою кількістю даних. Наприклад вони використовуються при оновленні перекладів, bulk операціях і т.д. Тобто ці маніпуляції виконуються у фоновому режимі без ручного втручання.

Batch операції налаштовані так, що вони можуть бути запущені до кінця, без людської взаємодії. Batch приймає набір файлів даних в якості вхідних даних, обробляє дані і виробляє набір файлів вихідних даних. Це операційне середовище називається як "пакетна обробка", оскільки вхідні дані зібрані в пакети або набори записів і кожна партія обробляється як єдине ціле, по кроково в залежності від навантаження сервера (завантаженості сервера).

4.7 Трансформація даних з програмного забезпечення медприладу до заданного вигляду у системі використовуючи таксонометричні словники

Таксономія («закон», «наука») – термін, що має кілька значень:

- наука про об'єднання живих істот у групи на основі аналізу притаманних їм ознак;
- групування речей або принципи, що лежать в основі цього групування.
- наука про принципи та способи класифікації й номенклатури складноорганізованих ієрархічних систем дійсності: органічного світу, об'єктів географії, геології, мовознавства, суспільства тощо. Синонім понять “систематика”, “класифікація”.

Завдання таксономії - визначення і теоретичне обґрунтування класифікаційних одиниць – таксонів, їх системи, супідрядності, співвідношення та обсягу. В кінці XX ст. виникла тенденція розуміти під таксономією розділ систематики, вчення про систему таксономічних категорій. [4].

У нашому випадку таксонометричний словник це одиниця збереження назви вхідних параметрів та їх станів.

4.8 Збереження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL

Перше що треба зрозуміти - це що таке нода. Нода -це елемент контенту, який ґрунтується на одній і тій же базовій структурі. Завдяки цьому нода може з легкістю розширятися, виглядає це як додавання до сторінок коментарів, вкладених файлів, опитувань, рейтингів і тд. Так само нода володіє базовими властивостями, які успадковуються будь-якого типу контенту.

Наступне поняття, яке необхідно засвоїти - це Поля. Контент сторінки складається з набору полів. Наприклад, заголовок сторінки або саме Body - все це поля. Завдяки тому що ми можемо додавати які завгодно поля, ми можемо створити просто неймовірні типи контенту.

Таким чином даний етап створюваного алгоритму буде реалізовуватися на основі вхідних даних типу матеріалу, з відповідними полями, що забезпечує формування Ноди і подальше її збереження у необхідному вигляді у СУБД MySQL.

4.9 Аналіз та трансформація даних

Це найважливіший етап в роботі програмного забезпечення, оскільки необхідно коректно опрацювати інформацію та зберегти її у зручному форматі для подальшого відображення.

Підчас аналізу проводиться перевірка на наявність необхідних даних у веб-інтерфейсі для коректного їх збереження. За відсутності необхідних елементів вони автоматично створюються системою за рахунок чітко продуманого і реалізованого програмного коду.

4.10 Створення користувачів

Кожний окремий набір даних про результати обстеження необхідно прив'язати до конкретного користувача. В кожному XML файлі першим параметром є інформація про пацієнта.

На основі цієї інформації відбувається перевірка наявності даного пацієнта у веб-інтерфейсі. За відсутності він буде автоматично створений системою.

4.11 Створення «візитів»

Після створення користувачів та необхідних елементів у веб-інтерфейсі варто підготувати та зберегти інформацію про результати обстеження даного пацієнта (відвідувача).

Для цього використовується структурна одиниця організації даних, нода, «візит», яка дозволяє організувати та зберегти коректним чином інформацію про обстеження пацієнта.

Для цього відбувається групування даних, приведення їх до заздалегідь погодженого формату та збереження всієї інформації у базі даних веб-інтерфейсу.

4.12 Генерація інтернет - інтерфейсу: графіки візиту, графіки динаміки, рекомендації

Веб-інтерфейс та мобільний додаток є основними елементами з якими будуть контактувати користувачі (пацієнти та лікарі) даного програмного комплексу.

Генерація інтернет - інтерфейсу проходить в декілька етапів.

Перший етап передбачає: отримання із бази даних необхідної інформації про певного пацієнта та його результатів обстеження. Після отримання даних, відбувається їх візуалізація приклад представлення якої відображено на рисунку 2.

Текстові дані відображаються в незмінному вигляді, а числові значення результатів обстеження представляються у вигляді стовбчастих графіків, що полегшує виявлення відхилень та встановлення правильного діагнозу

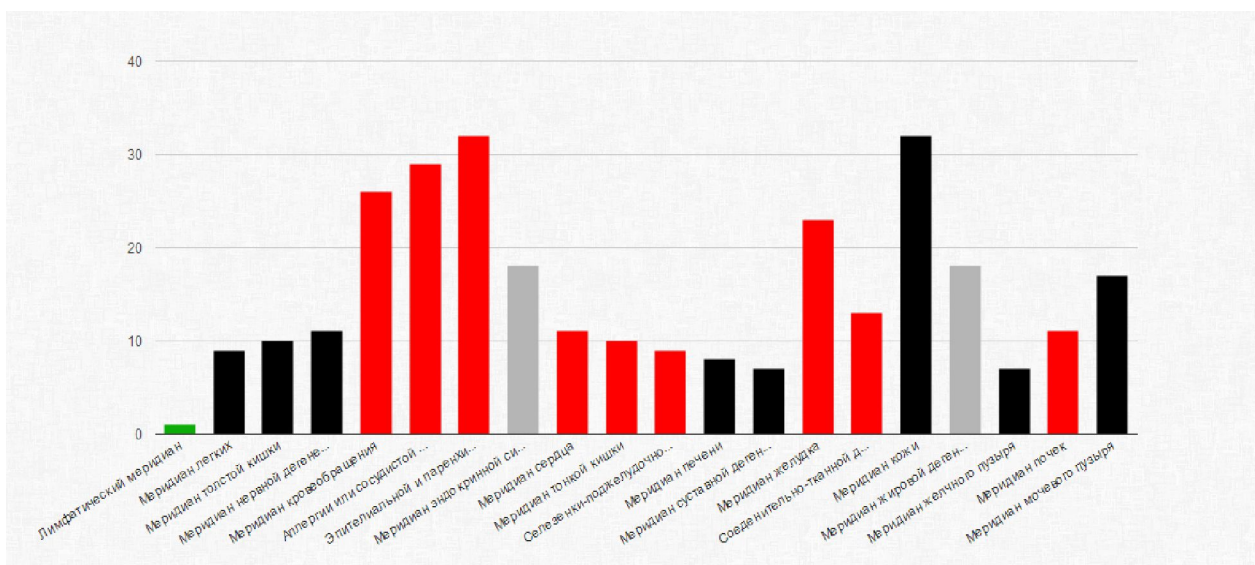


Рисунок 2. Приклад візуалізації отриманих даних

4.13 Мобільний Android додаток

Так як алгоритм роботи складається з генерації двох інтерфейсів web – інтерфейсу та інтерфейсу додатку то варто сказати декілька слів про роботу мобільного додатку.

Мобільний додаток працює при умові роботи веб-інтерфейсу. Нами був розроблений додаток для Android платформ під версією 2.2 і вище. Він в свою чергу використовує для відображення даних СУБД MySQL яка наповнюється інформацією від CMS Drupal.

На даному етапі розробки додаток може працювати тільки при активному підключенні до інтернету, так як не має власної БД. У майбутньому буде розроблена БД для мобільного додатку яка буде синхронізуватись з СУБД MySQL і тому не буде необхідності у постійному підключенні до інтернету.

Розробка є базовою тому аналогічний додаток може буди виконаний і для інших пристроїв, наприклад під керуванням IOS або Windows Phone.

Висновки

1) Таким чином розроблений нами алгоритм роботи системи трансформації даних від медприладів в web-інтрфейс складається з наступних кроків:

- завантаження архіву від програмного забезпечення мед-приладу;
- розархівування архіву та отримання його вмісту (XML файли);
- передача XML файлів в PHP обробники;
- обробка XML файлів перетворення вмісту в масив даних системи;
- запис батча та обхід масиву по елементно (парсинг) на основі розробленого функціоналу;
- трансформація даних з програмного забезпечення мед- апарату до вигляду у системі використовуючи аксонометричні словники;

- збереження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL, таблиця NODE;
- аналіз та трансформація даних;
- створення користувачів;
- створення «візитів»;
- генерація інтерфейсу;

2) Коротко описано розроблений та практично реалізований алгоритм роботи універсальної системи трансформації даних від мед-приладів у web-інтерфес. Алгоритм забезпечує високоефективне отримання даних із зовнішнього (локального) програмного продукту (мед-приладу) до обробки, збереження та подальшого їх відображення у в глобальній мережі інтернет в тому числі з можливістю відображення на мобільному додатку.

3) Розроблений мобільний Android додаток для Android платформ під версією 2.2 і вище.

Список використаної літератури

- 1) В.О. Сазик, О. М. Решетило, О.В. Сазик Обґрунтування вибору систем керування базою даних та управління контентом при створенні Web-сайту // Міжвузівський збірник «Комп'ютерно-інтегровані технології: освіта, наука, виробництво», 2013. - №12. - С.56-62.
- 2) http://www.open-classroom.net/php-lections/3578_vzaimodeiystviephpxml.html/
- 3) <http://www.saxproject.org/>

РОЗДІЛ 5

ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ РОБОТИ СИСТЕМИ ОТРИМАННЯ І ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ З ІНТЕРЕНТ ІНТЕРФЕЙСОМ НА СУБД MySQL

5.1 Завантаження архіва з XML файлами від програмного забезпечення медприладів

Для реалізації веб-форми завантаження архіву з XML файлами використано Drupal Form API, який представляє собою набір методів та засобів для роботи з веб-формами.

Код функції має наступний вигляд:

iron_magister_xml_load_form

за допомогою якої генерується елементи форми:

```
function iron_magister_xml_load_form($form, &$amp;form_state) {  
  $form = array();  
  $form['file_fid'] = array(  
    '#type' => 'managed_file',  
    '#title' => t('Please load folder with xml files'),  
    '#upload_validators' => array(  
      'file_validate_extensions' => array('zip'),  
    ),  
    '#upload_location' => 'public://xml/',  
    '#required' => TRUE,  
  );  
  $form['submit'] = array(  
    '#type' => 'submit',  
    '#value' => t('Load')  
  );  
  return $form;
```

}

Як видно із представлено PHP коду, створюється елемент *file_fid* який дозволяє завантажувати файли на сервер через веб-переглядач. Основними параметрами які вказуються при створені даного елемента веб-форми є:

- *file_validate_extensions* – даний параметр дозволяє вказувати які розширення файлів можна звантажувати, в даному випадку лише zip;
- *upload_location* – шлях до директорії на сервері куди буде збережено архів після завантаження;
- *submit* – елемента веб-форми, по натисканню на який відбувається завантаження архіву на сервер.

Для контролю процесу завантаження архіву реалізовано обробник. Приведемо код функції *iron_magister_xml_load_form_submit*, яка викликається по натисканню на елемент *submit*:

```
function iron_magister_xml_load_form_submit($form, &$form_state) {  
  $dir = drupal_realpath('public://xml/');  
  //Load the file.  
  $file = file_load($form_state['values']['file_fid']);  
  // Change status to permanent.  
  $file->status = FILE_STATUS_PERMANENT;  
  // Save.  
  $uploaded = file_save($file);  
  if ($uploaded == TRUE) {  
    drupal_set_message(t('The file has been uploaded.'));  
  }  
  else {  
    drupal_set_message(t('The file could not be uploaded. Please contact the  
site administrator.'), 'error');  
    return FALSE;  
  }  
  // Unpack the archive.  
  module_load_include('inc', 'update', 'update.manager');
```

```

    $archive = update_manager_archive_extract(drupal_realpath($file->uri),
$dir);

    // Delete the file archive, not to produce garbage on the server.
    file_delete($file);

    // Get a list of files from the archive.
    $xmlFiles = glob($dir . '/*.xml');
    // If there are files run import.
    if (!empty($xmlFiles)) iron_magister_import($xmlFiles);
    }

```

При активізації елементу *submit* відбувається завантаження файлу на сервер за допомогою функції *file_load*.

Після завантаження необхідно перемістити файл із тимчасової папки у постійну, це здійснюється за рахунок зміни статусу файла *\$file->status = FILE_STATUS_PERMANENT*; та виклику функції *file_save*, яка власне здійснює фізичне переміщення файлу.

Якщо файл не вдалося зберегти виводиться повідомлення про помилку, а у разі успішного збереження - про успіх операції.

5.2 Розархівування архіву та отримання його вмісту (XML файли)

Після збереження архіву відбувається його розархівування за допомогою функції *update_manager_archive_extract*, вхідними параметрами якої є шлях до архіву та шлях до папки в яку необхідно розархівувати архів. Після розархівування файл видаляється за допомогою функції *file_delete*.

5.3 Отримання вмісту архіву

Отримуємо список XML файлів, які знаходилися в архіві за допомогою функції *glob*. Програмна реалізація розглянутого нами в розділі 4 алгоритму роботи розробленої системи представлено на схемі 1.

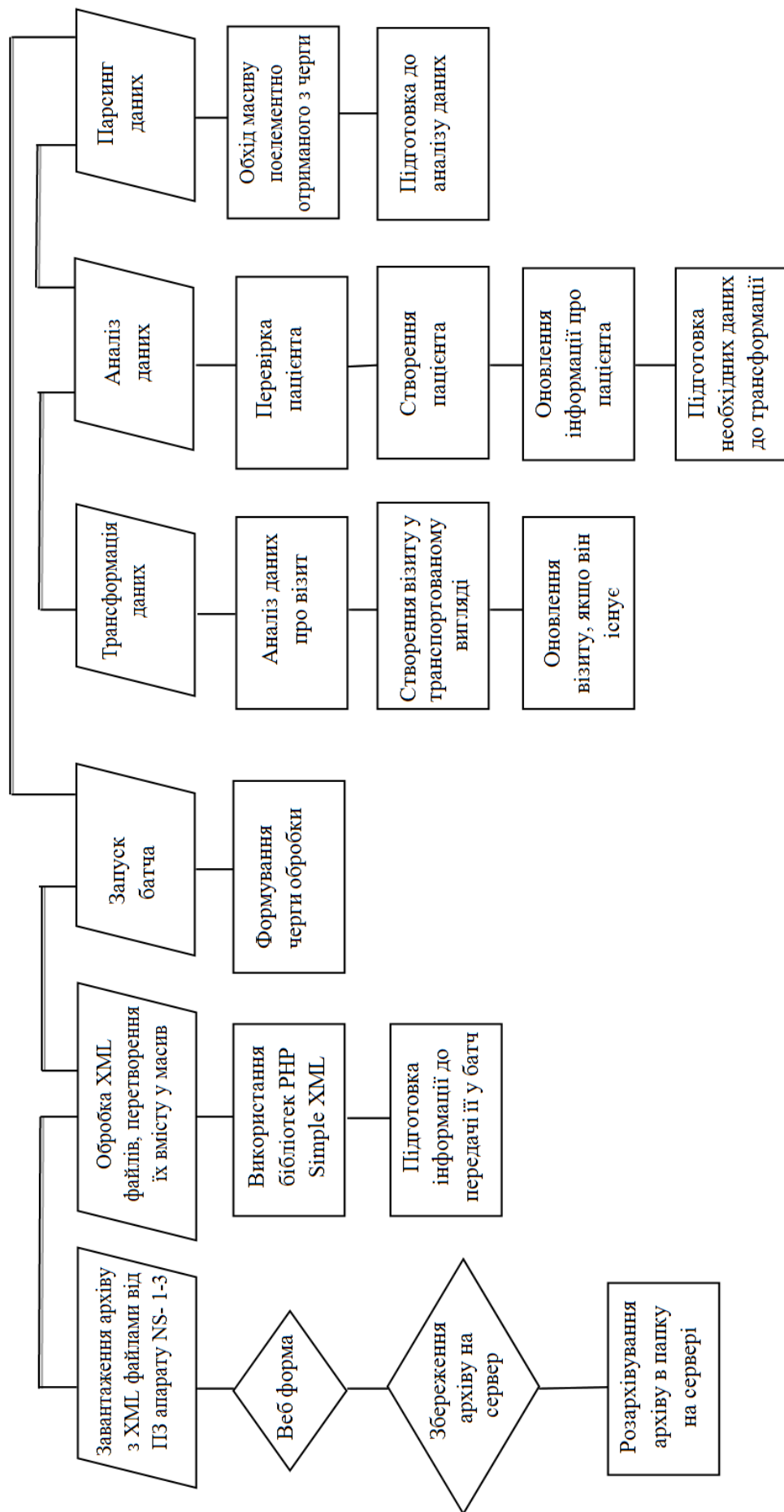


Схема 1. Програмна реалізація алгоритму роботи універсальної системи трансформації даних: медприлади – web – інтерфейс.

5.4 Обробка XML файлів, перетворення їх вмісту у масив даних системи

Якщо XML файли присутні у кінцевій директорії запускаємо процес обробки даних викликаючи функцію *iron_magister_import*, вхідним параметром для якої є масив шляхів до XML файлів.

Розглянемо код функції *iron_magister_import*:

```
function iron_magister_import($files) {  
    $batch = array(  
        'title' => t('Process'),  
        'init_message' => t('In progress'),  
        'progress_message' => t('Process data started'),  
        'error_message' => t('The processing is stopped due to an error!'),  
        'operations' => array(),  
        'finished' => 'IMbatch_end',  
        'file'      => drupal_get_path('module', 'iron_magister') .  
        '/inc/iron_magister.parser.inc',  
    );  
    $batch['operations'][] = array ('IMbatch_process', array($files));  
  
    batch_set($batch);  
    batch_process();  
  
    return TRUE;  
    }
```

5.5 Створення черги

В даній функції ініціалізується *Batch* з використанням *Drupal Batch API*, який дозволяє обробляти великі об'єми інформації, завдяки формування черги. *Batch* працює покроково, за кожен крок проводиться обробка заданої кількості інформації.

5.6 Запуск батча

Оператор *\$batch['operations'][] = array ('IMbatch_process', array(\$files));* дозволяє вказати основну функцію обробки та масив даних який необхідно обробити.

Розглянемо функцію *IMbatch_process*:

```
function IMbatch_process($files, &$context) {  
    if (empty($context['sandbox'])) {  
        $context['sandbox']['progress'] = 0;  
        $context['sandbox']['items'] = 0;  
        $context['sandbox']['max'] = count($files);  
    }  
  
    $limit = 5;  
  
    if(empty($context['sandbox']['items'])) {  
        $context['sandbox']['items'] = $files;  
    }  
  
    if(!empty($context['sandbox']['items'])) {  
        if ($context['sandbox']['progress'] != 0) {  
            array_splice($context['sandbox']['items'], 0, $limit);  
        }  
  
        foreach ($context['sandbox']['items'] as $element) {  
            if (@$counter != $limit) {  
                IMbatch_parser($element);  
  
                $counter++;  
  
                $context['sandbox']['progress']++;  
  
                $context['message'] = t('Now processing node %node of %count',  
array('%node' => $context['sandbox']['progress'], '%count' =>  
$context['sandbox']['max']));  
  
                $context['results']['nodes'] = $context['sandbox']['progress'];
```

```

        unlink($element);
    }
}

if ($context['sandbox']['progress'] != $context['sandbox']['max']) {
    $context['finished'] = $context['sandbox']['progress'] /
$context['sandbox']['max'];
}
}
}
}

```

В даній функції вказуються основні параметри обробки даних. А саме, *\$limit = 5*; вказує яку кількість елементів опрацьовувати за один крок, *IMbatch_parser(\$element)*; виклик функції, яка є основним аналізатор інформації, приймає єдиний вхідний параметр – шлях до XML- файлу.

5.7 Парсинг даних

Розглянемо функцію *IMbatch_parser*:

```

function IMbatch_parser($path) {

    // Read the information from the file.
    $dataXML = simplexml_load_file($path);

    // Check if the user is online with the given parameters.
    // If no such user - create it.
    $account = IMbatch_user_handler($dataXML->INFORMATION-
>USER);

    // Obtain information about the user's visit.

```

```

    $visit      =      IMbatch_visit_handler($dataXML->INFORMATION-
>SECTION, $account);

    // Having information about the user and the visit,
    // process data rezultath survey.
    IMbatch_survey_handler($dataXML, $visit, $account);
}

```

В даній функції відбувається зчитування інформації з XML-файла та передача її в PHP, за допомогою оператора *\$dataXML = simplexml_load_file(\$path);*.

5.8 Трансформація даних

Трансформація даних з програмного забезпечення медприладів до вигляду у системі та збереження даних у необхідному вигляді відбувається використовуючи тип матеріалу (створення користувачів, “візитів”).

Далі відбувається аналіз даних про пацієнта викликом функції *IMbatch_visit_handler*, котра перевіряє чи зареєстрований користувач з вхідними даними, якщо ні – то він буде створений.

Аналізуються дані про візит за допомогою функції *IMbatch_visit_handler*, яка перевіряє чи є у веб-інтерфейсі інформація про даний візит, якщо не має – він буде створений.

І останній етап це аналіз даних про результати обстеження пацієнта. Даний аналіз виконує функція *IMbatch_survey_handler*, яка підготує та трансформує до заздалегідь визначеного формату (суть якого буде більш детально розглядатись у наступному розділі) даних та збереження їх у базі даних веб-інтерфейсу.

5.9 Завантаження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL таблиця NODE та визначення стійкості роботи системи

Після закінчення етапу трансформації даних вони безпосередньо зберігаються у ноду окремого типу матеріалу в необхідному нам вигляді покрокова інструкція розкрита в розділі шість.

По факту функціонування системи стійкість її роботи можна визначити використовуючи аналітичні програми. У нашому випадку ми будемо використовувати *Google Analytics Tools* (рисунок 1).

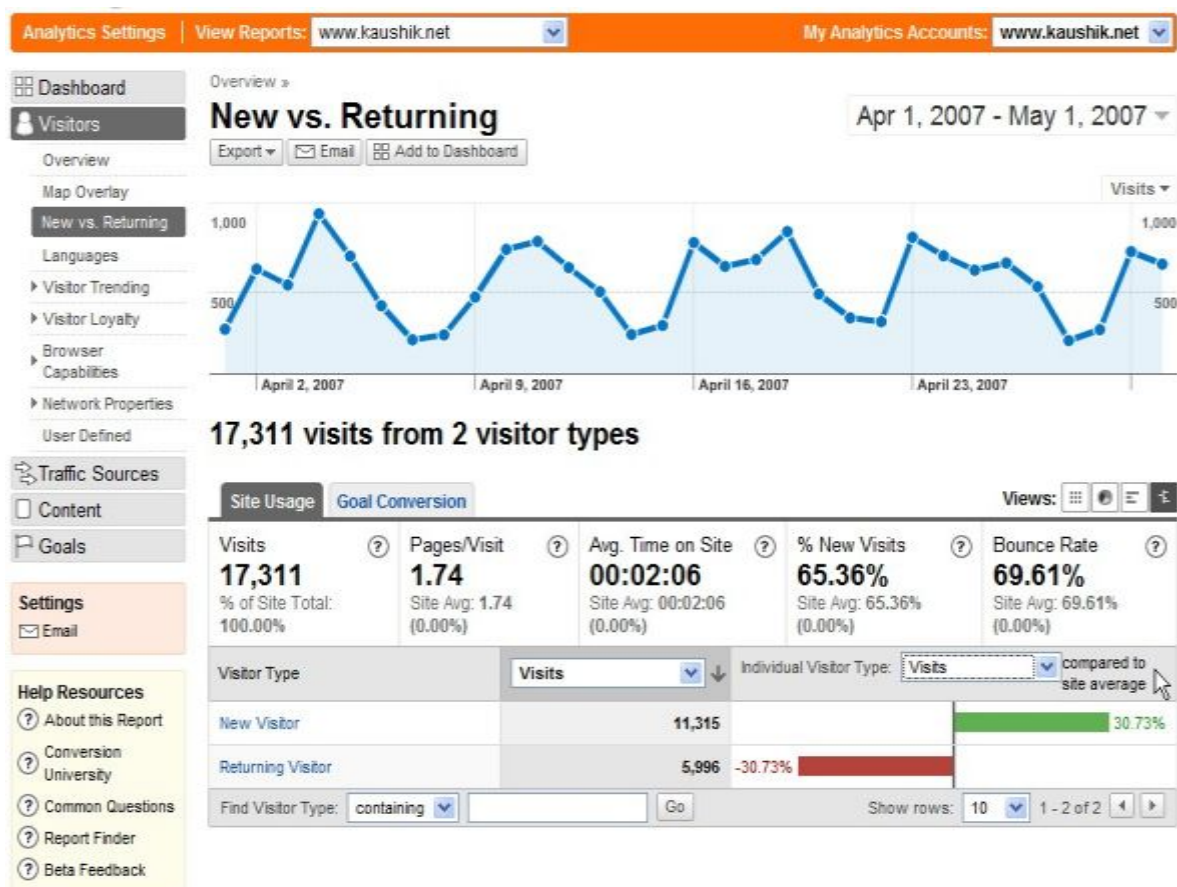


Рисунок 1. Визначення стійкості роботи системи використовуючи аналітичні Google Analytics Tools.

Завдяки даному доповненню ми зможемо проаналізувати кількість користувачів за певний період часу (що буде свідчити про успішність даного

веб - ресурсу), наявність і відсутність сторонніх атак та реакції системи на їх дії, завантаженість на сервер, слабкі сторони, стійкість системи при максимальному та мінімальному навантаженні.

Висновки

1) Для реалізації веб-форми завантаження архіву з XML файлами використано Drupal Form API, який представляє собою набір методів та засобів для роботи з веб-формами.

2) Розархівовування збережених даних відбувається за допомогою функції *update_manager_archive_extract*, вхідними параметрами якої є шлях до архіву та шлях до папки в яку необхідно видобути архів.

3) Список XML файлів, які знаходилися в архіві отримуємо за допомогою функції *glob*.

4) Обробка XML файлів, перетворення їх вмісту у масив даних системи здійснюється посередництвом запуску процесу обробки даних викликаючи функцію *iron_magister_import*, вхідним параметром для якої є масив шляхів до XML файлів.

5) Трансформація даних з програмного забезпечення медприладів до вигляду у системі та їх збереження даних відбувається використовуючи тип матеріалу (створення користувачів, “візитів”). Далі відбувається аналіз щодо наявності чи відсутності тих чи інших даних про результати обстеження пацієнта за допомогою функції *IMbatch_survey_handler*.

6) Програмна реалізація розробленого алгоритму забезпечує отримання даних від зовнішнього програмного забезпечення медприладів, трансформацію та їх збереження у веб-інтерфейсі системи на СУБД MySQL. Програмна реалізація виконана за допомогою серверної мови програмування PHP, використовуючи процедурний метод програмування.

7) Визначення стійкості роботи створеної системи здійснюється використовуючи аналітичні програмні продукти Google Analytics Tools.

Список використаної літератури

1) <http://www.google.com/analytics/>

РОЗДІЛ 6

ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОЇ СИСТЕМИ ТАНСФОРМАЦІЇ ДАНИХ ВІД МЕДПРИЛАДІВ У WEB ІНТЕРФЕЙС НА СУБД MYSQL

6.1 Вигляд даних у програмному забезпеченні діагностичного медприладу «ЗДОРОВ'Я»

Програмне забезпечення апарату отримує дані у вигляді чисельного значення електричного струму та конвертує ці значення у чисельно - буквене позначення. Можливі такі варіанти позначень (станів організму):

Н Н	норма;
С <- +> Н	метаболічна компенсація;
Н С	гомеостатична компенсація;
С <- +> С	початкові метаболічні порушення;
С <- +> П (1)	метаболічні порушення;
С <- +> П (2,3)	метаболічна патологія;
Н П (1)	стан декомпенсацій;
Н П (2,3)	виражена декомпенсація;
Ф (+, -) Н	дисфункція;
Ф (+, -) П (1,2,3)	патологічний процес.

- Позначення «Н» у першій колонках говорить про нормальну функцію;
- позначення «С» означає, що для даного елемента присутні початкові зміни функції за рахунок зниження адаптаційної активності зони;
- позначення «ф (+) >> - означає збудження;
- позначення «ф (-) >> - про пригноблення.

Кількість плюсів і мінусів, нестійке від 1-го до 3-х позначає ступінь зміни функції.

У другій колонці позначення «Н» говорить про нормальний стан

анатомічного субстрату, в поєднанні зі зміною функції - про нещодавно змінені процеси.

Позначення:

«П (1) >>»,

« П (2) >>

«П (3) >>

вказують на різні стадії порушення анатомічної будови органа.

Позначення «С» у другій колонці означає, що для даного елемента присутні початкові паталогічні зміни за рахунок зниження адаптаційної активності точки.

6.2 Вигляд даних у системі з інтенет-інтерфейсом на СУБД MYSQL

Для того, щоб у подальшому візуалізувати отримані дані у веб – інтерфейсі та у мобільному додатку, була вироблена система перетворень інформації яку ми отримуємо.

Для цього усі варіанти позначень було переведено у число та до кожного позначення ми додали колір.

Число 1 - норма. Збільшення цього числа говорить про порушення стану органу. Максимальне числове значення погіршення стану - 32 (чи то збудження, чи то пригнічення).

Збудження позначається червонім кольором. Пригнічення позначається чорним кольором. Інформація про стани зберігається у таксонометричному словнику.

В таблиці 6.1 зазначене числове значення стану органу в нашій системі, значення стану як у нашій системі так і у програмному забезпеченні медичного приладу, колір стану у нашій системі та значення стану органу у програмному забезпеченні медичного приладу.

6.3 Візуалізація даних

У створюваній нами системі, ми вирішили представити стан організму людини за окремими його органам у вигляді графіків. Тому для цього нами і була використана трансформація отриманих даних від програмного забезпечення медприладу у вигляді чисельно-буквенного позначення у просте чисельне позначення для можливості побудови будь яких схем, графіків чи діаграм, таблиця 1.

Зазначимо що вибраний нами метод трансформації був заздалегідь проаналізований і впроваджений виходячи з можливості використання уже трансформованих даних у будь яких математичних формулах, схемах, графіках чи діаграмах, що дозволить у подальшому розширити функціональність нашої системи.

Таблиця 1

Таблиця перетворень

Числове значення стану	Значення стану	Колір	Значення ПЗ апарату
1	Норма		Н Н
18	Стан декомпенсації		Н П[1]
19	Виражена декомпенсація 2-го рівня		Н П[2]
20	Виражена декомпенсація 3-го рівня		Н П[3]
14	Збудження адаптаційної активності зони 3-го рівня та метаболічна патологія 2-го рівня		С[+++] П[2]
17	Збудження адаптаційної активності зони 3-го рівня та метаболічна патологія 3-го рівня		С[+++] П[3]

13	Збудження адаптаційної активності зони 2-го рівня та метаболічна патологія 2-го рівня		C[++] П[2]
16	Збудження адаптаційної активності зони 2-го рівня та метаболічна патологія 3-го рівня		C[++] П[3]

Для побудови графіків з отриманих та трансформованих даних ми використали API (application programming interface) Google Chart. Сервіс як і все геніальне - простий: тепер все, що необхідно для побудови графіків це сформувати URL.

За допомогою даного сервісу ми легко зможемо будувати ефективні динамічні діаграми практично «на льоту».

Користуватись Google Chart API просто: всі параметри побудови діаграми і дані для візуалізації передаються на сервер Google у вигляді GET запиту, а сервер віддає вам PNG-зображення з намальованим графіком. Більше того, використання даного API дозволяє зняти з вашого сервера додаткове навантаження по генерації графіки та й за швидкодію не доводиться хвилюватися (так як Google має надпотужні сервери).

Доступні наступні типи діаграм:

Лінійна діаграма

Стовпчикова діаграма

Кругова діаграма

Діаграма Венна

Графік розсіювання

Приклад типів діаграм котрі підтримуються API (application programming interface) Google Chart представлено на рисунку 1.

Для кожного типу зображень можна вказувати атрибути, такі як розмір, кольори та підписи. Можна помістити зображення API Chart на веб-сторінку шляхом прописування URL для тегу . Коли веб-сторінка відображається в браузері, API Google Chart створює картинку і забезпечує її показ на сторінці.

Для побудови діаграм є детальна та вичерпна документація (на російській мові), яка дозволяє ознайомитися з принципами використання даного API і надає довідковий матеріал по доступних параметрах.

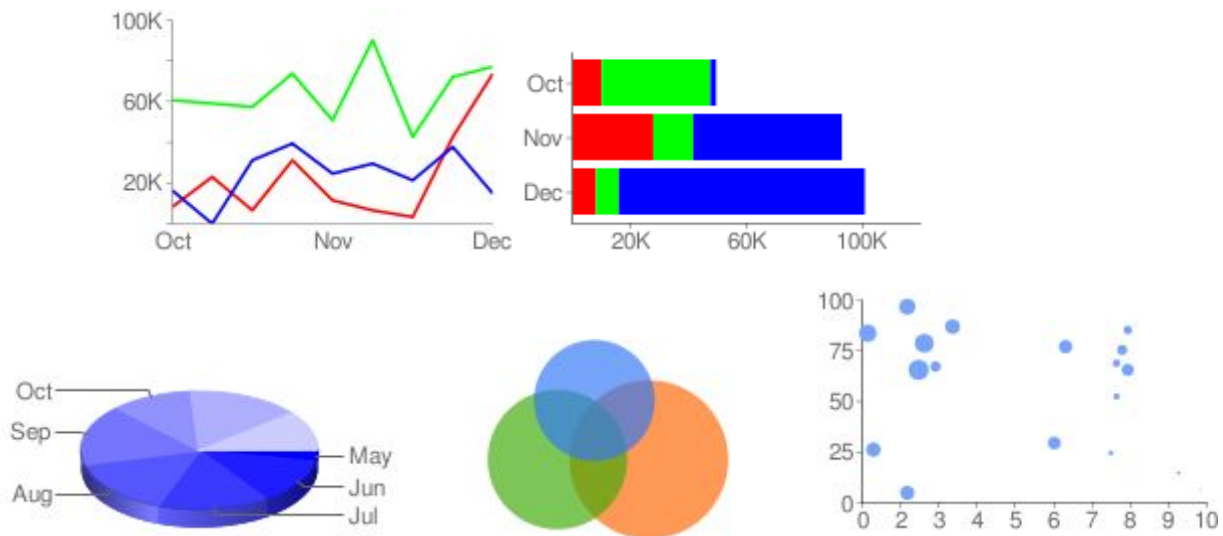


Рисунок 1. Приклади побудови діаграм Google Chart

6.4 Приклад синтезу PHP та JavaScript коду який виводить середньо – статистичні висновки про стан енергетичних меридіанів у вигляді графіка

Для побудови графіків представлених на рисунку 1, ми використали синтез серверної мови програмування PHP та JavaScript необхідний для роботи з Google Chart.

Нижче наведено приклад синтезу PHP та JavaScript коду який виводить середньо – статистичні висновки про стан енергетичних меридіанів у вигляді графіка:

```
/**
 * Charts in dynamic
 */
function iron_block_dynamic_charts() {

    global $user;
```

```
$html = '';
```

```
$word = 'Bizum';
```

```
// Робимо вибірку всіх візитів всіх користувачів
```

```
$q = db_select('node', 'd');
```

```
$q->condition('d.type', 'visit');
```

```
q->fields('d', array('nid'));
```

```
$res = $q->execute()->fetchAll();
```

```
// Підключаємо бібліотеку google chars api
```

```
$html .= '<script type="text/javascript"
```

```
src="https://www.google.com/jsapi"></script>';
```

```
$visits = array();
```

```
foreach($res as $key => $value){
```

```
$node = node_load($value->nid);
```

```
if($node->field_visit_patient['und'][0]['uid'] == arg(1) && arg(1) != 1){
```

```
$meridians_decode = json_decode($node-
```

```
>field_visit_meridians['und'][0]['value']);
```

```
$handler_state = array();$meridian_name = array();
```

```

foreach($meridians_decode as $keys => $val){

$meridian = taxonomy_term_load($val->meridian);

$meridian_name[] = "".$meridian->name.""";

$state = taxonomy_term_load($val->state);
$handler_state[] = $state->field_states_value['und'][0]['value'];
}
$visit_num = $node->field_visit_number['und'][0]['value'];

$visits[] = "[".$word . $visit_num . ""," . implode(',',$handler_state)."]";
}
}

if(@$meridian_name){$first =
"['"."Bizumu'," . implode(',',$meridian_name)."]";}

if(@$visits){$second = implode(',',$visits);}

$grafic = @$first . ' . ' . @$second;

// Виводимо графік
if (arg(1) && arg(1) > 1){
$html .= "<script type='text/javascript'>
google.load('visualization', '1', {packages: ['corechart']});
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = google.visualization.arrayToDataTable([
\".$grafic.\"
]);

```

```

var options = {
  title: '',
  backgroundColor: 'transparent',
  curveType: 'function'
};

var chart = new
google.visualization.LineChart(document.getElementById('dynamic_chart'));

chart.draw(data, options);
}
</script>";

$html .= '<h1>'.t('Зміни у динаміці').</h1>';

$html .= '<div class="dynamic-content">';

$html .= '<div class="content-grafics-title">Середньо - статистичні
висновки про стан енергетичних мережі</div><div id="dynamic_chart"
style="width: 100%; height: 800px;"></div>';

$html .= '</div>';
}

return $html;
}

```

6.5 Приклади створеного інтерфейсу системи отримання і збереження інформації на СУБД MySQL

Для входу у систему отримання і збереження інформації з інтернет – інтерфейсом на СУБД MySQL нами створена веб-форма, яку можна побачити на рисунку 2.

Інтерфейс розробленої системи складається з двох основних частин, основна - контентна частина та права бокова панель.



Рисунок 2. Сторінка входу в систему отримання і збереження інформації з інтернет – інтерфейсом на СУБД MySQL

В основній контентній частині відображено прізвище ім'я та по батькові відвідувача. В центрі розміщено графік зміни середньо-статистичних висновків про стан енергетичних меридіанів у динаміці по мірі використання рекомендацій лікаря та дотримання їх пацієнтом

Права частина бокова панель відображає сторінки візитів (їх на прикладі ми бачимо сім) та довідка – повідомлення пацієнту про дану динаміку. Дана довідка складається лікарем для розтлумачення стану роботи тих чи інших органів (меридіанів), рисунок 3.

BIG DATA

Середньо - статистичні висновки про стан енергетичних мереж

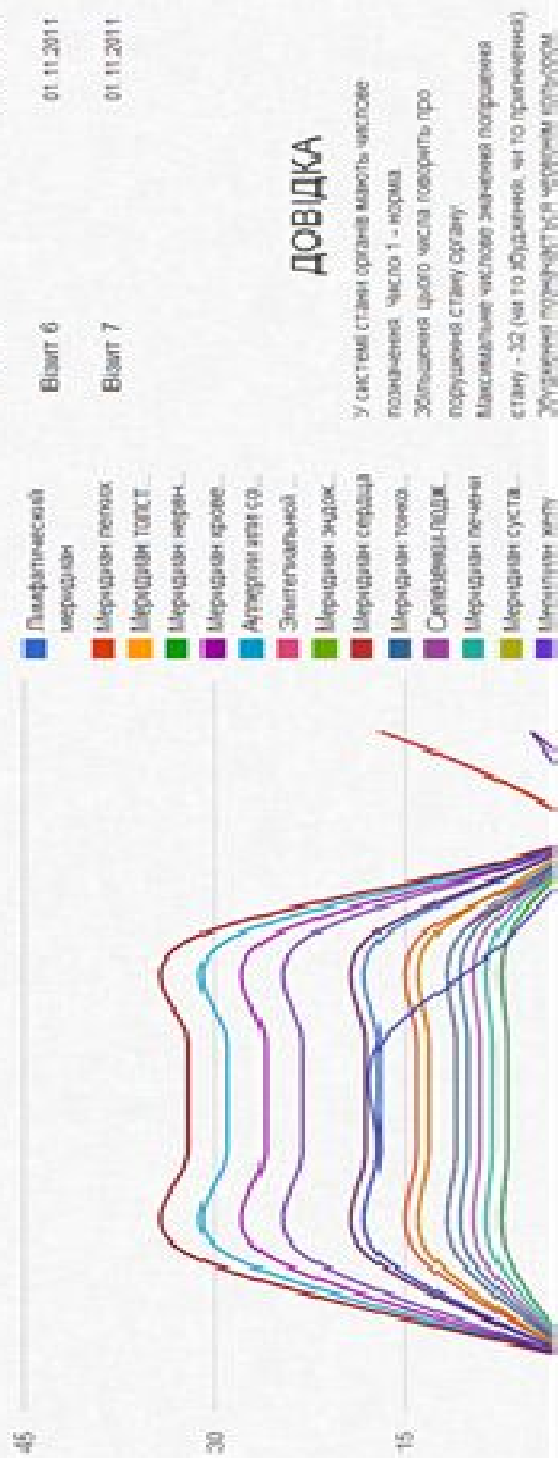


Рисунок 3. Інтерфейс системи отримання і збереження інформації з інтернет – інтерфейсом на СУБД MySQL для пацієнта

Система трансформації даних дозволяє створювати електронні картки пацієнтів, що є досить зручним у користуванні та підвищує рівень культури обслуговування. Загальний вигляд електронної картки відображена на рисунку 4.

З електронної картки лікар, пацієнт, зможе считати стан параметрів роботи окремих органів та загальний стан організму по окремому візиту, довідка, висновок і порада лікаря.

В основній контентній частині знаходяться побудовані нами графіки та інформація про пацієнта, а саме: ПП, дата народження, зріст та вага. У правій частині інформації про пацієнта розташовані дві кнопки для виходу з системи та переходу на головну сторінку де розташовані графіки у динаміці пацієнта.

У правій боковій панелі знаходиться набір блоків. Першим блоком йде меню, де відображено список візитів пацієнта. Наступним йде довідка яка допомагає зорієнтуватись у принципі роботи системи. Далі йдуть блоки в яких вказується інформація від лікаря, порада та висновок. Останнім йде блок з інформацією про стани елементів.

Інтерфейс для пацієнта і лікаря відрізняються.

Інтерфейс для лікаря: на місті списку візитів пацієнта, розміщено список всіх пацієнтів та їх візитів, з окремою кнопкою для редагування кожного візиту окремо, чи карточки пацієнта в цілому. Це можна побачити на рисунку 5.

На рисунку 6 відображена сторінка редагування візиту пацієнта та її наповнення, формування висновку та поради лікарем.

Доступ до сторінки редагування візиту пацієнта, її наповнення, формування висновку та формування поради лікаря має виключно приймаючий лікар.

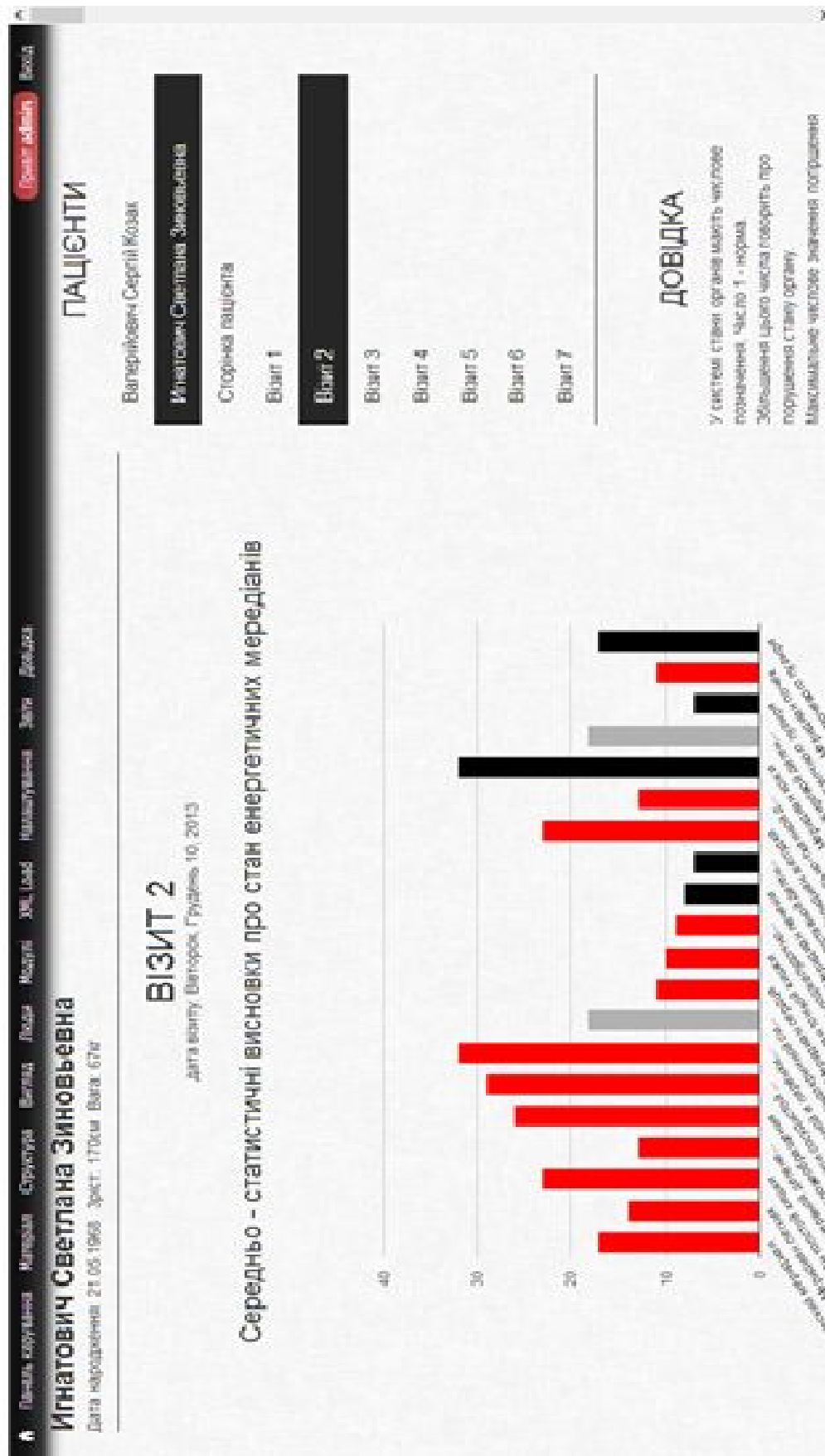


Рисунок 5. Інтерфейс системи отримання і збереження інформації з інтернет – інтерфейсом на СУБД MySQL для лікаря.

The screenshot shows a web application interface for editing patient visit information. The browser address bar displays the URL: `ironeggs-dev2.org.ua/node/7/edit?destination=admin/content`. The page contains several form fields:

- Title**: A text input field with the value "Phonetic, Castorina Lomaxova 2013".
- Patient**: A text input field with the value "p-1524 (at 30)".
- Number visit**: A text input field with the value "3".
- DATE VISIT**: A date picker field showing "12/12/2013".
- Medication**: A text area containing XML data for medications, including items like "medication 201", "medication 202", etc.
- Organs**: A text area containing XML data for organs, including items like "organ 210", "organ 211", etc.
- Conclusion physician**: A text area containing text in Ukrainian, starting with "Лікарський... це текст...".
- Advice to Physician**: A text area containing text in Ukrainian, starting with "Лікарський... це текст...".

Рисунок 6. Сторінка редагування візиту пацієнта

Доступ до сторінки редагування інформації про пацієнта має виключно адміністратор (лікар). Дану сторінку умовно можна розділити на дві важливих частини.

До першої частини входять активаційні дані для авторизації у системі які ми формуємо на основі даних отриманих від програмного забезпечення медприладу. Активаційні дані містять: прізвище, ім'я по батькові, адреса електронної пошти, пароль, налаштування мови.

До другої умовної частини включено: прізвище, ім'я, по батькові, ідентифікатор пацієнта, дата народження, зріст, вага, висновок та порада лікаря. Варто відмітити що прізвище, ім'я та по батькові, ідентифікатор пацієнта, дата народження, зріст, вага формуються за допомогою розробленого функціоналу при обробці XML файлів.

Висновок та порада лікаря (загальні) формуються лікарем виходячи з аналізу стану параметрів підсистем організму людини у динаміці та виходячи з досвіду та міркувань лікаря.

6.6 Обґрунтування необхідності створення мобільного додатку

Мобільний додаток працює при умові роботи веб-інтерфейсу. Нами був розроблений додаток для Android платформ під версією 2.2 і вище. Він в свою чергу використовує для відображення даних СУБД MySQL яка наповнюється інформацією від CMS Drupal. На даний момент додаток може працювати тільки при активному підключенні до інтернету, так як поки що не має власної БД. У майбутньому буде розроблена БД для мобільного додатку яка буде синхронізуватись з СУБД MySQL. І тому необхідність у постійному підключенні до інтернету відпаде сама собою.

Розробка є базовою тому аналогічний додаток може бути виконаний і для інших пристроїв, наприклад під керуванням iOS або Windows Phone.

Додаток дозволяє забезпечити додатковий і найчастіше самий стабільний і постійний канал для взаємодії з пацієнтами. Можна виділити декілька основних моментів у необхідності створення додатка.

По-перше, зручність. Відвідування звичайних сайтів і в цілому вихід в інтернет з компактних кишенькових пристроїв часто супроводжується незручністю, а особливо - втратою часу і грошей. Вони занадто часто йдуть на завантаження тих додаткових (але не завжди необхідних) мультимедійних елементів, якими насичені віртуальні ресурси. Мобільні додатки завантажуються значно швидше і може працювати навіть без підключення до мережі, тобто оффлайн (у подальшому). У цьому можна виділити не один, а відразу кілька плюсів: економія часу і коштів, можливість використання додатків у відсутності зв'язку або коштів на рахунку (у подальшому), швидкодія (за рахунок фільтрації інформації та задіяння внутрішніх резервів пристрою).

По-друге, всі мобільні додатки спеціально створюються під конкретну модель пристрою. Це дає можливість не тільки максимально синхронізувати їх роботу, а й зробити її максимально результативною. Сюди можна віднести і ті

оновлення, які регулярно створюються і періодично скидаються на додатки, навіть незалежно від того, підключені вони до мережі на даний конкретний момент чи ні (якщо зв'язку немає, поновлення просто чекатимуть її відновлення).

6.7 Середовище розробки

Для розробки мобільного додатку було вибрано Eclipse IDE середовище.

Eclipse - вільне інтегроване середовище розробки модульних кроссплатформених додатків. Розвивається і підтримується Eclipse Foundation.

Найбільш важливими особливостями Eclipse є:

- кроссплатформеність Eclipse виконується на всіх поширених платформах: Windows, Linux і MacOS X. Ще важливіше те, що його функції однакові на кожній з цих платформ;
- універсальність і розширюваність: у Eclipse реалізована можливість використання різних інструментів, розроблених «сторонніми» програмістами;
- відкритість і безкоштовність: Eclipse є OpenSource - проектом (тобто його вихідні коди доступні будь-якому охочому і хто завгодно може приєднатися до розробки цього інструменту). Eclipse має активне співтовариство, постійно працює над поліпшенням програми і розширенням її можливостей.

6.8 Мова програмування мобільного додатку

Для розробки мобільних додатків на Android пристрої використовується мова програмування Java.

Java - об'єктно-орієнтована мова програмування. Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури.

Перевагою подібного способу виконання програм є повна незалежність байт - коду від операційної системи і устаткування, що дозволяє виконувати Java -додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Основні переваги об'єктно-орієнтованої мови програмування Java:

- переносимість, або кросплатформенність;
- об'єктна орієнтованість, створена ефективна об'єктна модель;
- звичний синтаксис C / C + +;
- вбудована і прозора модель безпеки;
- орієнтація на Internet-завдання, мережеві розподілені додатки;
- динамічність, легкість розвитку і додавання нових можливостей;
- простота освоєння.

Приклад програмної реалізації мобільного додатку представлено на схемі 1.

6.9 XML розмітка

<?xml version="1.0" encoding="UTF-8"?>

<manifest android:versionName="1.0" android:versionCode="1"
package="com.example.ironmagister"

xmlns:android="http://schemas.android.com/apk/res/android"><uses-sdk

android:targetSdkVersion="8" android:minSdkVersion="8"/><uses-

permission android:name="android.permission.INTERNET"/><application
android:theme="@style/AppTheme" android:label="@string/app_name"

android:icon="@drawable/ic_launcher"

android:allowBackup="true"><activity

android:name="com.example.ironmagister.UserPage"

android:label="@string/app_name"><intent-filter><action

android:name="android.intent.action.MAIN"/><category

android:name="android.intent.category.LAUNCHER"/></intent-filter></activity></application>

</manifest>

6.10 JAVA код

package com.example.ironmagister;

import android.annotation.SuppressLint;

import android.app.Activity;

import android.os.Bundle;

import android.webkit.WebView;

public class UserPage extends Activity {

@SuppressLint("SetJavaScriptEnabled")

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.user_page);

WebView myWebView = (WebView) findViewById(R.id.webview);

myWebView.getSettings().setJavaScriptEnabled(true);

myWebView.getSettings().setLoadsImagesAutomatically(true);

myWebView.clearCache(true);

}

}

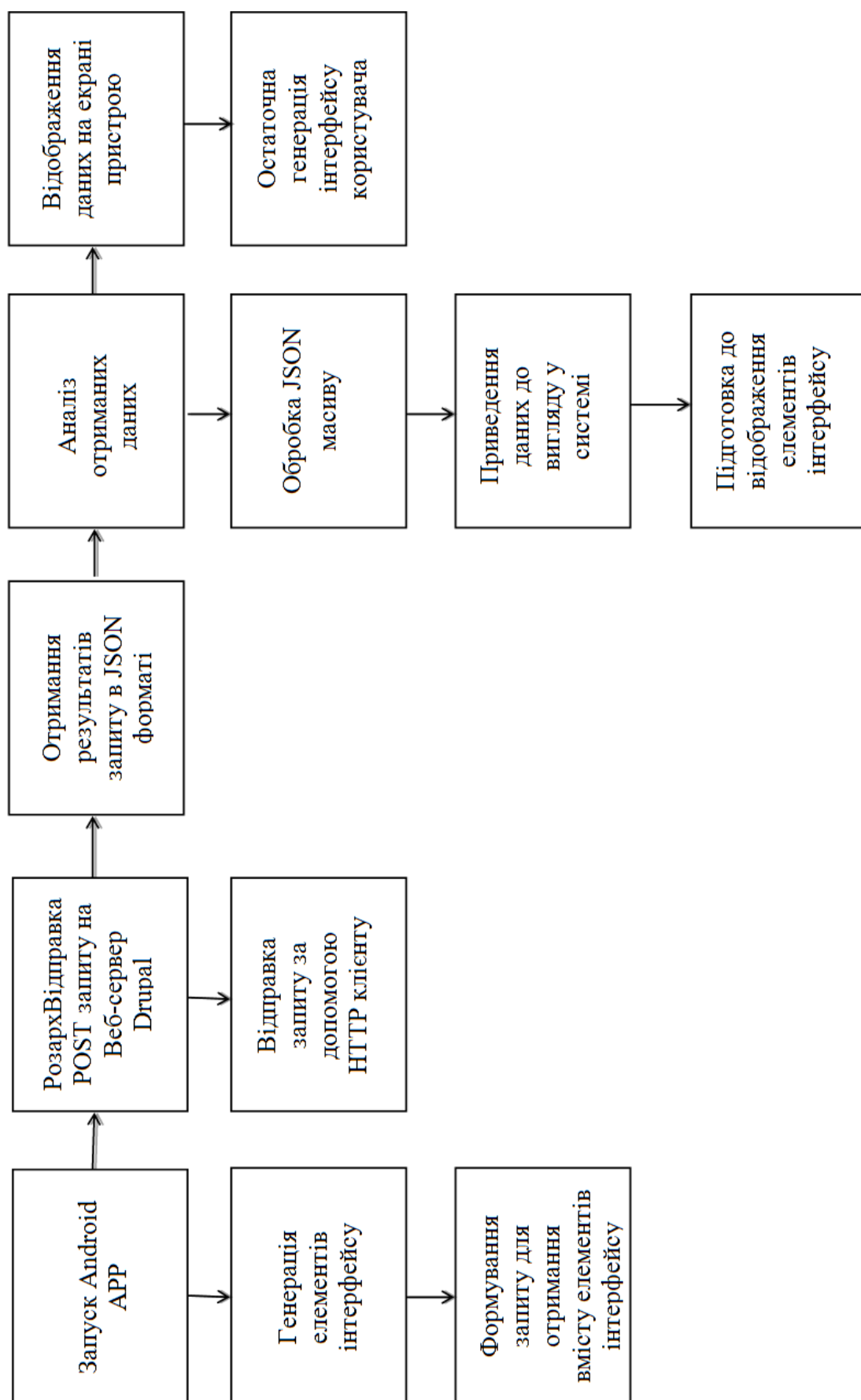


Схема 1. Приклад програмної реалізації мобільного додатку

Висновки

- 1) Програмне забезпечення апарату отримує дані у вигляді чисельного значення електричного струму та конвертує ці значення у чисельно - буквене позначення.
- 2) З метою візуалізації отриманих даних у веб – інтерфейсі та у мобільному додатку, була вироблена система перетворень для цього всі варіанти позначень було переведено у число та відповідний колір.
- 3) Загальний стан організму людини та за окремими його органами візуалізуються в web-інтрефейсі у вигляді графіків, на основі трансформації отриманих даних від програмного забезпечення мед - приладу у вигляд чисельно-буквенного позначення в просте чисельне позначення для можливості побудови будь-яких схем, графіків та діаграм.
- 4) Інтерфейс розробленої системи складається з двох основних частин: контентної та правої бокової панелі, кожна з яких наповнена відповідною інформацією.
- 5) Права частина бокова панель відображає сторінки візитів (їх на прикладі ми бачимо сім) та довідка – повідомення пацієнту про дану динаміку.
- 6) Проведено обґрунтування та розроблено додаток для Android платформ під версією 2.2 і вище. Додаток дозволяє забезпечити додатковий і найчастіше самий стабільний і постійний канал для взаємодії з пацієнтами.
- 7) Для розробки мобільного додатку було вибрано Eclipse IDE - вільне інтегроване середовище розробки модульних кроссплатформенних додатків.
- 8) Для розробки мобільних додатків на Android пристрої використовується мова програмування Java.

- 9) Проаналізувавши виконане даного розділу, можна з впевненістю сказати що розроблена досить складну з точки зору реалізації систему отримання та збереження інформації з інтернет інтерфейсом на СУБД MySQL. Крім того ми впровадили чи малу кількість трансформацій даних для подальшої їх візуалізації багатьма способами.
- 10) У процесі розробки та написання програмного коду керувались принципом універсальності, тому наша система є базовою для подальшого її розвитку, гнучкою для внесення змін та впровадження нововведень.

Список використаної літератури

- 1) <http://vispyanskiy.name/>

РОЗДІЛ 7

ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБКИ СИСТЕМИ ТРАНСФОРМАЦІЇ ДАНИХ ВІД МЕДПРИЛАДІВ У WEB –ІНТРЕФЕЙС

Економічна ефективність є важливою складовою при розробці будь-якої системи. Проведемо перелік проведення необхідних робіт для розробки системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL*. Визначимо час необхідний для проведення відповідних робіт (таблиця 1).

Таблиця 1

Найменування етапів розробки та тривалості їх реалізації

Номер п/п	Найменування етапів	Тривалість етапів год.	Тривалість етапів %
1	Підготовчий етап	16	4,4
2	Розробка теоретичної частини теми	80	21,7
3	Розробка системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL	160	43,5
4	Розробка мобільного додатку	80	21,7
5	Внесення коректив у розроблену систему	24	6,5
6	Завершальний етап	8	2,2
	Всього	368	100

Для вирахування вартості розробленої системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL нами був проведений аналіз ринку розробки аналогічних систем.

З урахуванням складності виконаної нами роботи, її реалізація може бути виконана тільки Senior PHP розробником. Найменша ціна яка встановлюється для такого рівня програмування за час роботи становить 8 євро.

Вартість розроблюваної системи на даний момент можна вирахувати за формулою:

$$\text{summ} = w * k; \quad (8.1)$$

де: summ – сума у грошовому еквіваленті в євро валюті,

w – кількість затрачених годин,

k – ставка за час роботи.

Підставляємо дані із таблиці 8.1 у приведену вище формулу (8.1):

$$160 \text{ год.} * 8 \text{ євро} = 1280 \text{ євро.}$$

Таким чином отримуємо, що вартість проведення робіт по розробці системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL (пункт 3 табл.8.1) становить 1280 євро.

Для розрахунку вартості розробки мобільного додатку, нами також був виконаний аналогічний аналіз ринку. Для виконання робіт такого рівня найменша вартість години роботи JAVA розробника під Android платформи дорівнює 12 євро. Формула використовується ідентична попередній.

Підставляємо відповідні дані:

$$80 \text{ год.} * 12 \text{ євро} = 960 \text{ євро}$$

Таким чином вартість розробки мобільного додатку (пункт 4 табл.8.1) становить 960 євро.

Що стосовно теоретичної частини, то при розрахунках вартості виконуваних робіт, ми за основу обрали мінімальну заробітну плату на

підприємствах по місту, котра складає 1600 грн. Визначаємо вартість роботи години працівника за наступною формулою:

$$\text{count} = \text{zp} / (\text{d} * \text{h}); \quad (8.2)$$

де count – грошовий еквівалент години роботи працівника у грн.,

zp – заробітна плата працівника,

d – кількість робочих днів у місяці,

h – кількість робочих годин в одному робочому дні.

Підставляємо відповідні дані:

$$1600 / (20 * 8) = 10 \text{грн} = 1 \text{євро}$$

Таким чином, вартість роботи години працівника в розробці теоретичної частини становить 1 євро.

Для підрахування вартості усіх виконаних теоретичних робіт використаєм формулу:

$$\text{summ} = \text{w} * \text{k}; \quad (8.3)$$

Підставляємо відповідні дані:

$$128 \text{год.} * 1 \text{євро} = 128 \text{євро.}$$

Таким чином, загальна вартість теоретичних робіт становить 128 євро.

Для підрахування загальної суми витрат додамо її складові (затрати: на теоретичну частину, розробку системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL та розробку мобільного додатку):

$$1280 + 960 + 128 = 2368 \text{євро}$$

Таким чином, виходячи з усіх підрахунків можна зробити висновок, що для розробки системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL необхідно затратити близько 24000 гривень.

Розроблена нами система отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL на етапі розвитку буде розповсюджуватись безкоштовно. Але для забезпечення стабільної роботи сервера на якому буде

функціонувати наша система та збільшення її функціональних можливостей необхідні відповідні витрати на її обслуговування. За для забезпечення надходження коштів передбачається використання реклами на сайті зокрема Google AdWords.

Як показує аналіз подібних методів, маємо високу ймовірність отримання прибутку таким чином. Вирішувальним фактом буде велика кількість візитів користувачів на сайт. Зокрема це підтверджує рисунок 1.

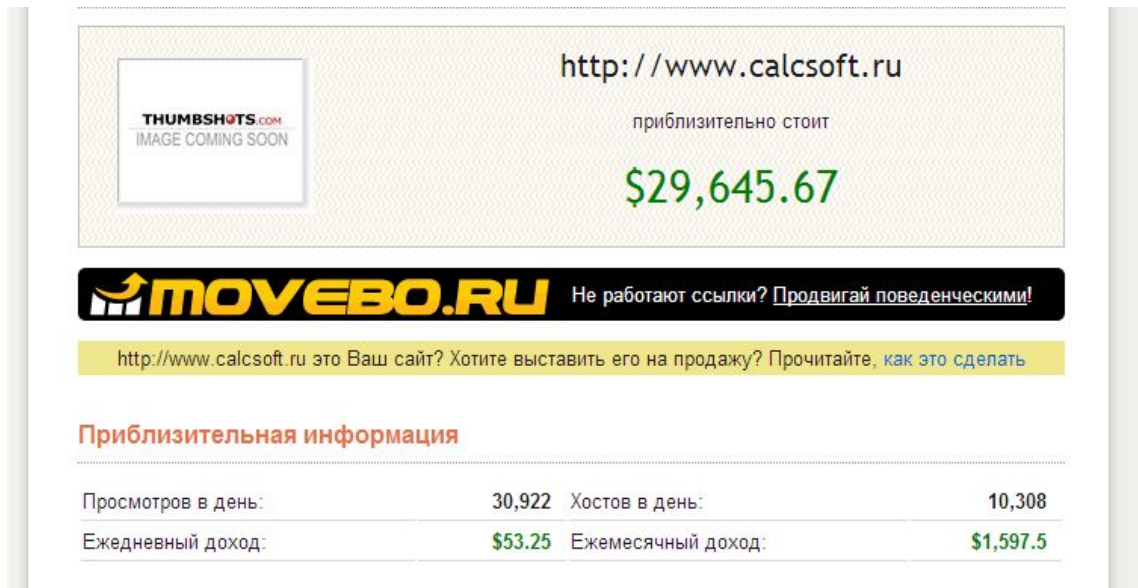


Рисунок 1. Приклад прибутку від реклами.

На рисунку 1 бачимо сайт з системою кредитних калькуляторів <http://calcsoft.ru/>, зображено прибуток сайту від реклами з невеликою кількістю користувачів (усього близько 30000).

Висновок

Таким чином, загальна вартість розробки системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL становить близько 24000 гривень.

**Для розрахунку економічної ефективності за основу на базові об'єкти були взяті ціни станом на 10.01.2014 року.*

Список використаної літератури

- 1) me.kmu.gov.ua/file/link/86954/file/Metod.doc
- 2) <http://calcsoft.ru/>,

ЗАГАЛЬНІ ВИСНОВКИ

1) Таким чином в даному розділі ми проаналізували різні визначення поняття система та зупинилися на наступному: система - ціленаправлена взаємодія елементів за заданою шкалою цінностей.

2) Проводячи класифікацію систем ми виділяємо відповідно три найбільших типи систем: костні (мертві), живі та соціальні, всі інші типи можна віднести до цих трьох.

3) З нашої точки зору найбільш прикладими є наступні властивості системи: синергетика та емергентність.

4) Синергетика – процес самоорганізації котрий відбувається як в костних, живих так і в соціальних системах. Суть синергетики в ході розвитку невірноважених процесів при деякому критичному значенні зовнішнього потоку енергії або речовини із-за втрати стійкості із неупорядкованих можуть виникати упорядковані стани.

5) Емергентність - незведеність окремих властивостей системи котрі проявляє система в цілому до властивостей окремих її елементів.

6) Медичний прилад – це технічний пристрій, що призначений для спостереження, вимірювання, обробки і представлення інформації про живий організм та для передачі цієї інформації на відстань.

7) Що стосується системи керування реляційними базами при створенні універсальної системи трансформації даних: медприлади – web інтерфейс з економічної точки зору, простоти програмування, підтримки та ефективності її роботи найбільш доцільно використовувати систему керування реляційними базами даних MySQL.

8) Зробивши аналіз найпоширеніших систем керування вмістом, ми вибираємо CMS Drupal, тому що, для розробки нашої системи дана CMS має необхідну достатність елементів та високий рівень комунікації між цими елементами.

Необхідна достатність елементів CMS Drupal включає в себе:

- кастомізовані типи матеріалу;
- таксонометричні словники;
- модульну систему;
- розширену роботу з формами;
- великий набір стандартних функцій для оптимізації.

9) Структура бази полів MySQL у CMS Drupal на наш погляд є досить зрозумілою та комфортною у використанні.

10) В MySQL визначено три базових типи даних: числовий, дата і час, рядковий.

11) Використовуваний нами популярний веб - сервер баз даних MySQL, за умови правильного вибору типів полів, забезпечує оптимізацію передачі даних в web-інтерфейс.

12) Числові дані обробляються сервером баз даних завжди швидше, в порівнянні із символьними, тому всі поля, в яких передбачається зберігати цифрову інформацію, слід створити з оголошенням відповідного типу: INT, FLOAT і так далі.

13) Ще на етапі проектування баз даних здійснюється обґрунтований вибір типів стовпців. Варто використовувати типи, які займають менше місця. Дві головні задачі, вирішення яких забезпечить оптимізацію роботи сервера з базою даних - це швидкість і об'єм займаного дискового простору.

14) Визначений діапазон можливих значень поля і заданий їх тип забезпечують мінімальну зайнятість об'ємного дискового простору та збільшення продуктивності роботи MySQL.

15) Власне програмно-апаратний комплекс ЗДОРОВ'Я складається з 2-х частин:

- апаратного забезпечення;
- програмного забезпечення.

В свою чергу апаратне забезпечення включає в себе:

- приставка 1 шт: (аналоговий пристрій для прийняття показників електрошкірного опору акупунктурної точки);
- комп'ютер (ноутбук) з приладдям 1 шт;
- принтер з приладдям 1 шт.;
- стандартний кабель USB;
- кабель електродів;
- щуп з кнопкою і трубчастий загальний електрод 1 шт.

16) Принцип роботи давачів параметрів фізіологічного стану людини: залежність величини електрошкірного опору акупунктурної точки від впливу нормованого струму, результати якого обробляються програмним забезпеченням побудованим за методом Голубцова В.В.

17) Програмне забезпечення апарату розроблене на базі об'єктно орієнтованої мови програмування DELPHI. Воно включає в себе дві частини:

- графічна оболонка (інтерфейс лікаря);
- системи об'єктів (кнопок) на які написані обробники для виконання тих чи інших дій: обробка та збереження даних в архів по візиту пацієнта у XML форматі без, на жаль, подальшої передачі даних у зовнішні системи.

18) На етапі доводки розроблюваної системи отримання і збереження інформації з інтернет інтерфейсом на СУБД MySQL розроблено функціонали по передачі архіву з XML файлами у зовнішні системи від програмного забезпечення апарату.

19) Розроблений нами алгоритм роботи системи трансформації даних від медприладів у web-інтрфейс складається з наступних кроків:

- завантаження архіву від програмного забезпечення мед-приладу;
- розархівування архіву та отримання його вмісту (XML файли);
- передача XML файлів в PHP обробники;
- обробка XML файлів перетворення вмісту в масив даних системи;
- запис батча та обхід масиву по елементно (парсинг) на основі розробленого функціоналу;
- трансформація даних з програмного забезпечення медапарату до

вигляду у системі використовуючи аксонометричні словники;

- збереження даних у необхідному вигляді використовуючи тип матеріалу в СУБД MySQL, таблиця NODE;
- аналіз та трансформація даних;
- створення користувачів;
- створення «візитів»;
- генерація інтерфейсу;

20) Коротко описано розроблений та практично реалізований алгоритм роботи універсальної системи трансформації даних від мед-приладів у web-інтерфес. Алгоритм забезпечує високоефективне отримання даних із зовнішнього (локального) програмного продукту (мед-приладу) до обробки, збереження та подальшого їх відображення у в глобальній мережі інтернет в тому числі з можливістю відображення на мобільному додатку.

21) Розроблений мобільний Android додаток для Android платформ під версією 2.2 і вище.

22) Для реалізації веб-форми завантаження архіву з XML файлами використано Drupal Form API, який представляє собою набір методів та засобів для роботи з веб-формами.

23) Розархівовування збережених даних відбувається за допомогою функції *update_manager_archive_extract*, вхідними параметрами якої є шлях до архіву та шлях до папки в яку необхідно видобути архів.

24) Список XML файлів, які знаходилися в архіві отримуємо за допомогою функції *glob*.

25) Обробка XML файлів, перетворення їх вмісту у масив даних системи здійснюється посередництвом запуску процесу обробки даних викликаючи функцію *iron_magister_import*, вхідним параметром для якої є масив шляхів до XML файлів.

26) Трансформація даних з програмного забезпечення медприладів до вигляду у системі та їх збереження даних відбувається використовуючи тип матеріалу (створення користувачів, “візитів”). Далі відбувається аналіз щодо

наявності чи відсутності тих чи інших даних про результати обстеження пацієнта за допомогою функції *IMbatch_survey_handler*.

27) Програмна реалізація розробленого алгоритму забезпечує отримання даних від зовнішнього програмного забезпечення медприладів, трансформацію та їх збереження у веб-інтерфейсі системи на СУБД MySQL. Програмна реалізація виконана за допомогою серверної мови програмування PHP, використовуючи процедурний метод програмування.

28) Визначення стійкості роботи створеної системи по здійснюється використовуючи аналітичні програмні продукти Google Analytics Tools.

29) Програмне забезпечення апарату отримує дані у вигляді чисельного значення електричного струму та конвертує ці значення у чисельно - буквене позначення.

30) З метою візуалізації отриманих даних у веб – інтерфейсі та у мобільному додатку, була вироблена система перетворень для цього всі варіанти позначень було переведено у число та відповідний колір.

31) Загальний стан організму людини та за окремими його органами візуалізуються в веб-інтерфейсі у вигляді графіків, на основі трансформації отриманих даних від програмного забезпечення мед - приладу у вигляд чисельно-буквенного позначення в просте чисельне позначення для можливості побудови будь-яких схем, графіків та діаграм.

32) Інтерфейс розробленої системи складається з двох основних частин: контентної та правої бокової панелі, кожна з яких наповнена відповідною інформацією.

33) Права частина (бокова панель) відображає сторінки візитів (їх на прикладі ми бачимо сім) та довідка – повідомлення пацієнту про дану динаміку.

34) Проведено обґрунтування та розроблено додаток для Android платформ під версією 2.2 і вище. Додаток дозволяє забезпечити додатковий і найчастіше самий стабільний і постійний канал для взаємодії з пацієнтами.

35) Для розробки мобільного додатку було вибрано Eclipse IDE - вільне інтегроване середовище розробки модульних кроссплатформених додатків.

36) Для розробки мобільних додатків на Android пристрої використовується мова програмування Java.

37) Розроблена досить складна з точки зору реалізації система отримання та збереження інформації з інтернет інтерфейсом на СУБД MySQL, котра забезпечує передачу даних від різних медприладів в веб-інтрефейс. Крім того ми впровадили чималу кількість трансформацій даних для подальшої їх візуалізації багатьма способами.

38) У процесі розробки та написання програмного коду керувались принципом універсальності, тому наша система є базовою для подальшого її розвитку, гнучкою для внесення змін та впровадження нововведень.

Для нотаток

Для нотаток

Наукове видання

Сацик Віктор Олександрович
Гарлінський Артем Ігорович
Сомов Дмитро Олександрович

**Універсальна система
трансформації даних:
мєдприлади – web-інтерфейс**

Монографія

Друкується в авторській редакції
Технічний редактор *В. О. Сацик*

Формат 60x84 1/16. Обсяг 6,74 ум. друк. арк., 6,58 обл.-вид. арк.
Наклад 100 пр. Зам. 635. Видавець і виготовлювач – Вежа-Друк
(м. Луцьк, вул. Бойка, 1, тел. (0332) 29-90-65).
Свідоцтво Держ. комітету телебачення та радіомовлення України
ДК № 4607 від 30.08.2013 р.