

УДК: 621.301

## РОЗРОБКА ПЛАТФОРМИ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ДЛЯ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ВІДКРИТИМ КОДОМ OPENEMR

**А. В. Семенець, В. П. Марценюк**

*ДВНЗ «Тернопільський державний медичний університет  
імені І. Я. Горбачевського МОЗ України»*

Показано актуальність використання та альтернативні підходи до впровадження МІС в галузі охорони здоров'я України. Проаналізовано можливості застосування СППР в при діагностиці патологій невиношування вагітності. Наведено результати розробки платформи СППР як модуля (плагіна) для вільно-розповсюджаної МІС ЕМК з відкритим кодом OpenEMR. Розроблено інформаційну модель БД СППР. Реалізовано діалоговий компонент СППР з використанням API, наявного в МІС OpenEMR. Розроблено адміністративний модуль СППР з використанням PHP-фреймворку Yii2. Показано підходи до реалізації алгоритму процесу прийняття рішення у вигляді окремого сервісу засобами GoogleAppEngine.

**Ключові слова:** медичні інформаційні системи, електронні медичні картки, системи підтримки прийняття рішень, дерева рішень, програмне забезпечення з відкритим кодом, МІС, ЕМК, СППР, OpenEMR, GoogleAppEngine, Yii2.

## РАЗРАБОТКА ПЛАТФОРМЫ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ ДЛЯ МЕДИЦИНСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ОТКРЫТЫМ КОДОМ OPENEMR

**А. В. Семенец, В. П. Марценюк**

*ГВУЗ «Тернопольский государственный медицинский университет  
имени И. Я. Горбачевского МЗ Украины»*

Показана актуальность использования и альтернативные подходы к внедрению МИС в области здравоохранения Украины. Проанализированы возможности применения СППР в при диагностике патологий невынашивания беременности. Приведены результаты разработки платформы СППР как модуля (плагина) для свободно-распространяемой МИС ЭМК с открытым кодом OpenEMR. Разработана информационная модель БД СППР. Реализован диалоговый компонент СППР с использованием API, имеющегося в МИС OpenEMR. Разработан административный модуль СППР с использованием PHP-фреймворка Yii2. Показано подходы к реализации алгоритма процесса принятия решения в виде отдельного сервиса средствами GoogleAppEngine.

**Ключевые слова:** медицинские информационные системы, электронные медицинские карты, системы поддержки принятия решений, деревья решений, программное обеспечение с открытым кодом, МИС, ЭМК, СППР, OpenEMR, GoogleAppEngine, Yii2.

## ON THE CDSS PLATFORM DEVELOPMENT FOR THE OPEN-SOURCE MIS OPENEMR

**A. V. Semenets, V. P. Martsenyuk**

*SHEI «Ternopil State Medical University by I. Ya. Horbachevsky of MPH of Ukraine»*

The importance and alternative approaches to the implementation of the MIS in the Ukraine healthcare system are discussed. The CDSS capabilities in diagnostic of the pregnancy pathologies are analyzed. The results of the CDSS platform development as the plugin for the open-source MIS OpenEMR are presented. The information model of the CDSS database was proposed and developed. The CDSS dialog subsystem implementation according to the MIS OpenEMR programming API is shown. The administrative module of the CDSS platform based on the Yii2 php-framework is created.

An approach to the decision making algorithm implementation based on the separate service which was complete with GoogleAppEngine is proposed

**Key words:** medical information systems electronic medical records, decision support systems, decision tree, open-source software, MIS, EMR, CDSS, OpenEMR, GoogleAppEngine, Yii2.

**Вступ.** Ключовою ланкою в інформатизації галузі охорони здоров'я України є широке запровадження медичних інформаційних систем (МІС). Швидкий прогрес у сфері інформаційних технологій дає практичну можливість підвищити якість надання медичної допомоги населенню шляхом забезпечення персоналу лікувальних закладів технічними засобами для ефективної обробки клінічної інформації [1]. Ефективність лікарської допомоги суттєво залежить від наявності повної та достовірності інформації як про поточний стан здоров'я пацієнта, так і про попередні клінічні випадки. Формування та підтримка електронної медичної картки пацієнта (EMR – Electronic Medical Record, ЕМК) є одним з концептуальних напрямків впровадження сучасних інформаційних технологій у лікувальному закладі [1–3].

## 1. Застосування МІС з відкритим кодом

### 1.1. Ринок МІС з відкритим кодом

Світовий ринок МІС демонструє динамічне зростання вже протягом багатьох років [3, 4]. В роботі [3] наведено огляд підходів до впровадження МІС та наведено короткий перелік провідних розробників МІС різного типу. Українськими виробниками програмного забезпечення також створено ряд якісних МІС, наприклад, «Доктор Елекс» (<http://www.doctor.eleks.com>), «EMCiMED» (<http://www.mcmed.ua>), «Каштан» (<http://www.ciet.kiev.ua>) та інші. Однак вказані МІС є комерційним ПЗ, з відповідними основними недоліками:

- висока вартість, що сягає сотень тисяч гривень;
- закритість програмного коду, який є власністю компанії-розробника;
- залежність від компанії-розробника у питаннях сервісного обслуговування МІС.

Паралельно з ринком комерційного ПЗ активно розвивається напрям застосування в галузі охорони здоров'я вільно-розповсюдженого ПЗ з відкритим кодом [1, 5]. Для прикладу, Вікіпедія наводить перелік відкритого ПЗ МІС, що містить більше 20 найменувань [6]. Широко застосовуються такі МІС ЕМК з відкритим кодом як WorldVista (<http://worldvista.org/>), OpenEMR (<http://www.open-emr.org/>) та OpenMRS (<http://openmrs.org/>) [1, 7]. По-

зитивними рисами ПЗ МІС з відкритим кодом від провідних світових розробників є [3]:

- власне безкоштовний характер як самої ПЗ МІС так і додаткових компонентів, таких як ПЗ сервера баз даних (БД) чи веб-сервера;
- широкі функціональні можливості, що часто не поступаються комерційним ПЗ МІС;
- крос-платформеність – більшість ПЗ МІС з відкритим кодом є веб-орієнтованими додатками і можуть працювати на будь-якій платформі, включно з сучасними мобільними пристроями;
- доступність вихідних кодів та програмних інтерфейсів для створення власних доповнень.

Вказані переваги роблять вільно-розповсюджене ПЗ МІС, і, в першу чергу, ПЗ МІС ЕМК, особливо актуальним для використання в країнах, що розвиваються, або у країнах, що перебувають в умовах жорсткої фінансової кризи – як Україна. Перспективи використання вільно-розповсюдженого ПЗ МІС в країнах, що розвиваються, детально розглянуто в роботах F. Aminpour, F. Fritz, C. J. Reynolds та інших [1, 3, 5, 7].

На жаль, в Україні не приділяється достатньо уваги перспективам застосування вільно-розповсюдженого ПЗ МІС. Мають місце лише поодинокі експериментальні розробки. Зокрема, на кафедрі медичної інформатики Тернопільського державного медичного університету імені І. Я. Горбачевського (ТДМУ) досліджуються підходи до впровадження МІС OpenMRS та розробляються методики інтеграції даної МІС ЕМК з іншим ПЗ МІС [3, 8].

### 1.2. Застосування систем підтримки прийняття рішення (CDSS) в акушерстві

Впровадження МІС не повинно обмежуватися лише ПЗ МІС ЕМК. Застосування в щоденній практичній діяльності сучасного лікаря клінічних систем підтримки прийняття рішення (CDSS – Clinical Decision Support Systems, СППР) є необхідною умовою підвищення якості надання медичної допомоги. В роботах [9–11] підтверджено позитивну залежність між процесом впровадження СППР та підвищенням якості медичної допомоги. Переваги від застосування СППР в лікувальних закладах країн, що розвиваються, показані в ро-

боті [12]. Важливість інтеграції МІС різного типу, насамперед МІС СППР та МІС ЕМК, досліджено в роботі [13].

Слід зазначити, що науковцями кафедри медичної інформатики ТДМУ протягом тривалого часу розробляються як теоретичні засади, так і програмні засоби СППР, що відображено в цілому ряді робіт, зокрема [14–18].

Одним з перспективних напрямків застосування СППР є діагностика патологій невиношування вагітності та запобігання передчасним пологам. Методи застосування, як МІС СППР зокрема, так і МІС ЕМК загалом, для раннього виявлення патологій невиношування вагітності в акушерстві та гінекології аналізуються в роботах багатьох авторів [19–21]. В ТДМУ Жилиєв М. М., Жилиєв М. І., Геряк С. М., Васильєва Н. А. досліджують фактори, що впливають на патологічний перебіг і спричиняють невиношування вагітності [22–25]. У 2013 році Семенець А. В., Жилиєв М. М. та Геряк С. М. розробили прототип СППР діагностики патологій невиношування вагітності [26]. Дана СППР реалізує модель вірогіднісного (статистичного) підходу в процесі прийняття діагностичного рішення [14, 26]. Позитивні результати дослідної експлуатації вищевказаного прототипу СППР в Тернопільському обласному перинатальному центрі «Мати і дитина» протягом 2013–2014 років дали підставу для розробки повнофункційного додатку МІС СППР діагностики патологій невиношування вагітності [27].

**Мета роботи** – представлення досвіду авторів з розробки модуля (плагіна) для вільно-розповсюджуваної МІС ЕМК OpenEMR, що реалізує функціональні можливості платформи СППР (CDSS).

## **2. Реалізація процесу розробки платформи системи підтримки прийняття рішення**

### **2.1. Засоби OpenEMR щодо розробки доповнень**

МІС OpenEMR включає ряд програмних засобів, що дають можливість користувачам системи розробляти власні модулі (плагіни) доповнень:

1. Інтерактивний редактор для створення шаблонів форм для реєстрації даних огляду пацієнта – Layout Based Visit Forms ([http://www.open-emr.org/wiki/index.php/LBV\\_Forms](http://www.open-emr.org/wiki/index.php/LBV_Forms));

2. Візуальний редактор для створення шаблонів формулювання медичних записів – Nation Notes ([http://www.open-emr.org/wiki/index.php/Nation\\_Notes](http://www.open-emr.org/wiki/index.php/Nation_Notes));

3. Набір програмних інтерфейсів (API) для створення користувацьких модулів (плагінів) на мові програмування PHP ([http://www.open-emr.org/wiki/index.php/The\\_OpenEMR\\_API](http://www.open-emr.org/wiki/index.php/The_OpenEMR_API)).

Інструменти 1 та 2 є простими у використанні та не потребують знання мов програмування. Однак, вони не містять засобів, необхідних для реалізації функціоналу СППР. Тому для розробки модуля (плагіна), що реалізує функціональні можливості платформи СППР, автори використали програмний інтерфейс для розробки форм (діалогових компонентів) OpenEMR ([http://www.open-emr.org/wiki/index.php/The\\_Forms\\_API](http://www.open-emr.org/wiki/index.php/The_Forms_API)). При цьому необхідне володіння мовами програмування HTML, PHP та SQL.

### **2.2. Розробка інформаційної моделі CDSS плагіна**

Підставою для розробки інформаційної моделі СППР стала робота [25], що містить необхідні числові дані для реалізації вірогіднісного підходу в процесі прийняття діагностичного рішення. Представлення інформаційної моделі СППР, для цього випадку, та необхідні структури даних, відповідають описаним в роботах [16, 18]. Одночасно автори застосували альтернативний метод моделювання процесу прийняття діагностичних рішень на основі реалізації алгоритму побудови «дерева рішень», з використанням технології DataMining, як описано в роботах [15, 17, 18]. Такий підхід дає можливість підвищити достовірність кінцевого діагностичного рішення. Одночасно є можливість здійснення оцінки якості вказаних алгоритмів прийняття рішення.

На етапі розробки прототипу СППР діагностики патологій невиношування вагітності [26] внутрішнє представлення інформаційної моделі було реалізовано у вигляді БД засобами системи управління реляційними базами даних (СУРБД) Microsoft Access (рис. 1).

На наступному етапі авторами було прийнято рішення про розробку універсального додатку, що реалізує функціональні можливості платформи СППР в складі вільно-розповсюджуваної МІС OpenEMR. При цьому внутрішнє представлення інформаційної моделі було модернізоване та адаптоване для роботи у складі БД МІС OpenEMR під управлінням СУРБД MySQL (рис. 2).

Роботу з даними пацієнта в БД реалізовано з використанням MVC-підходу (MVC, Model-View-Controller). При цьому для реалізації процесу анкетування пацієнта розроблено такі класи моделі:

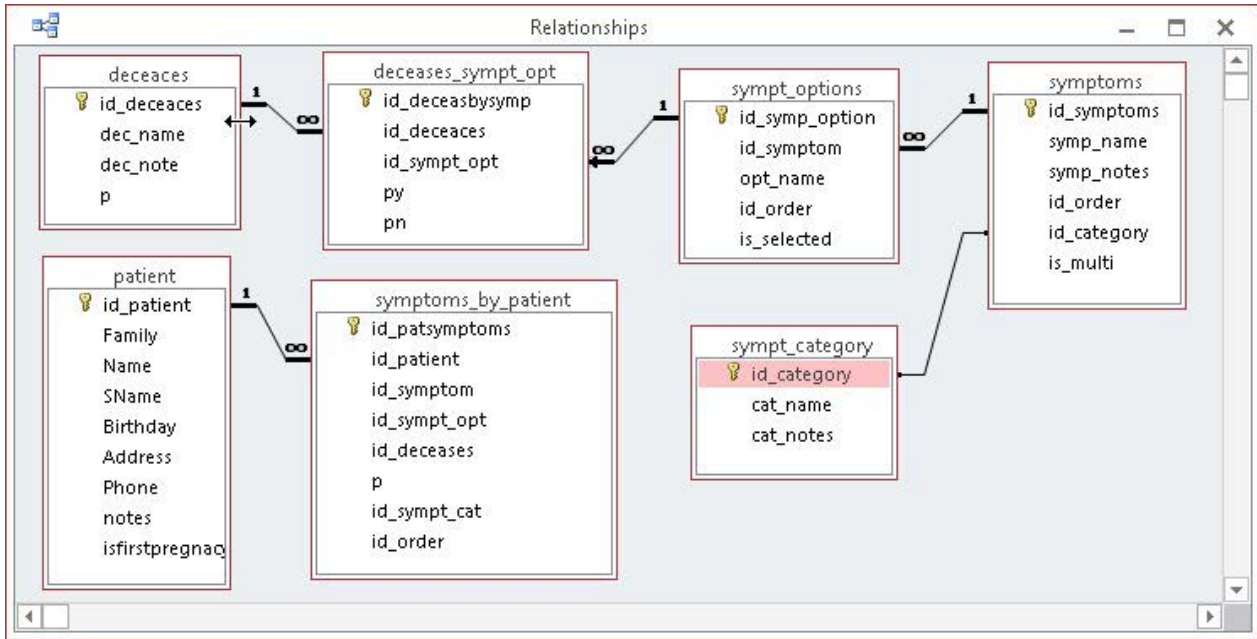


Рис. 1. Структура БД прототипу СППР діагностики патологій невиношування вагітності (СУРБД Microsoft Access).

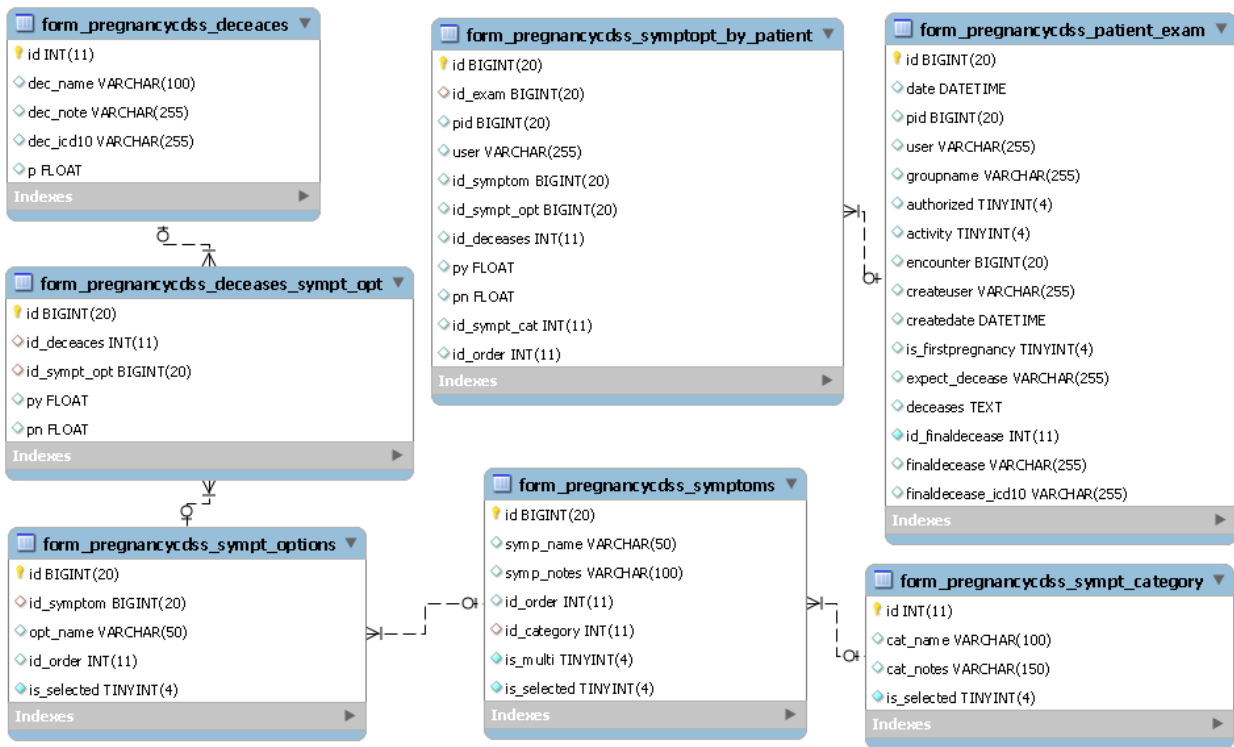


Рис. 2. Структура таблиць БД СППР діагностики патологій невиношування вагітності у складі БД МІС OpenEMR(СУРБД MySQL).

- SymptCategory\_Model.class.php – модель категорії симптомів;
- Symptoms\_Model.class.php – модель набору симптомів;

- SymptOptions\_Model.class.php – модель значень симптомів;
- DeceasesSymptOpt\_Model.class.php – модель вірогідності певного діагнозу залежно від значення симптому;

– SymptByPatient\_Model.class.php – модель результатів анкетування пацієнта.

Зокрема, в лістингу (лістинг 1) наведено реалізацію моделі результатів анкетування пацієнта у вигляді класу на мові програмування PHP. У відповідності до вимог МІС OpenEMR щодо розробки користувацьких додатків ([http://www.open-emr.org/wiki/index.php/Development\\_Policies](http://www.open-emr.org/wiki/index.php/Development_Policies)), робота з БД здійснюється з використанням бібліотеки ADODB (<http://adodb.sourceforge.net/>).

**Лістинг 1. Клас моделі результатів анкетування пацієнта**

```
class SymptByPatient_Model {
    var $id;
    var $id_exam;
    var $pid;
    var $user;
    var $id_symptom;
    var $id_sympt_opt;
    var $id_deceases;
    var $p_val;
    var $id_sympt_cat;
    var $id_order;
    public function __construct($id,
    $id_exam, $pid, $user, $id_symptom,
    $id_sympt_opt, $id_deceases, $py, $pn,
    $id_sympt_cat, $id_order) {
        $this->id = $id;
        $this->id_exam = $id_exam;
        $this->pid = $pid;
        $this->user = $user;
        $this->id_symptom = $id_symptom;
        $this->id_sympt_opt = $id_sympt_opt;
        $this->id_deceases = $id_deceases;
        $this->py = $py;
        $this->pn = $pn;
        $this->id_sympt_cat = $id_sympt_cat;
        $this->id_order = $id_order;
    }
    public static function all() {
        $list = [];
        $db = get_db();
        $req = $db->Execute('SELECT * FROM '.
SYMPTBYPATIENT_DBTABLE);
        foreach($req as $symptbypt) {
            $list[$symptbypt['id']]
= new SymptByPatient_Model
($symptbypt['id'], $symptbypt
['id_exam'], $symptbypt['pid'],
$symptbypt['user'], $symptbypt['id_
```

```
symptom'], $symptbypt['id_
sympt_opt'], $symptbypt['id_
deceases'], $symptbypt['py'],
$symptbypt['pn'], $symptbypt['id_
sympt_cat'], $symptbypt['id_order'],
$symptbypt['content']);
        }
        return $list;
    }
    public static function find($form_
idexam) {
        $list = [];
        $form_idexam = intval($form_idexam);
        $db = get_db();
        $req = $db->Execute('SELECT * FROM '.
SYMPTBYPATIENT_DBTABLE.' WHERE id_exam =
'.$form_idexam.' ORDER BY id_sympt_cat,
id_symptom, id_order');
        // the query was prepared, now we
replace :id_category with our actual
$id_category value
        foreach($req as $symptbypt) {
            $list[$symptbypt['id']] = new
SymptByPatient_Model ($symptbypt['id'],
$symptbypt['id_exam'],
$symptbypt['pid'], $symptbypt['user'],
$symptbypt['id_symptom'],
$symptbypt['id_sympt_opt'], $symptbypt
['id_deceases'], $symptbypt['py'],
$symptbypt['pn'], $symptbypt['id_
sympt_cat'], $symptbypt['id_order'],
$symptbypt['content']);
        }
        return $list;
    }
    //Check is it symptom option is selected
in given form for given patient
    public static function isselected($form_
idexam, $pid, $id_symptom, $id_symp_
option) {
        unset($req);
        $form_idexam = intval($form_idexam);
        $pid = intval($pid);
        $id_symp_option = intval($id_symp_
option);
        $id_symptom = intval($id_symptom);
        $db = get_db();
        $req = $db->Execute('SELECT * FROM
'.SYMPTBYPATIENT_DBTABLE.' WHERE (id_exam
= '.$form_idexam.')AND(pid = '.$pid.')
AND(id_sympt_opt = '.$id_symp_option.)
```

```

AND(id_symptom = '.$id_symptom.));
if ($req->RecordCount(>0){
return TRUE;
}else{
return False;
}
}
public static function selected
OptionsCount($form_idexam, $pid, $id_
symptom) {
unset($req);
$form_idexam = intval($form_idexam);
$pid = intval($pid);
$id_symptom = intval($id_symptom);
$db = get_db();
$req = $db->Execute('SELECT * FROM '.
SYMPTBYPATIENT_DBTABLE.' WHERE (id_exam
= '.$form_idexam.)AND(pid = '.$pid.)
AND(id_symptom = '.$id_symptom.));
return $req->RecordCount();
}
}

```

### 2.3. Реалізація діалогового компонента CDSS плагіна

При розробці користувацьких додатків для MC OpenEMR є можливість застосовувати більшість сучасних підходів до веб-програмування, включно з технологіями HTML5, CSS3, AJAX (переважно, засобами JQuery). Для СППР діагностики патологій невиношування вагітності у складі MIC OpenEMR авторами розроблено основну форму анкети, що реалізує діалоговий компонент СППР, а також ряд форм-звітів. Вказані форми, в рамках застосування MVC-підходу, реалізовано у вигляді таких представлень:

- SymptByPatient2\_Form.html – основна форма анкети (рис. 3);
- SymptByPatient\_Form2Report.class.php – клас представлення даних СППР діагностики патологій невиношування вагітності у формі звертання пацієнта у MIC OpenEMR (рис. 4);
- SymptByPatient\_Form2Print.class.php – клас представлення протоколу анкетування пацієнта та діагностичного висновку (рис. 5).

Рис. 3. Діалоговий компонент СППР діагностики патологій невиношування вагітності у складі MIC OpenEMR.

Пациєнт: **Наталія Іваненко (2)**  
Дата / нар.: 1994-04-16 Вік: 21

Хронологія звернень  
Обране звернення: **2015-04-04 (2)**

**Іваненко, Наталія** (Видалити)

Хронологія | Звіт | Документи | Транзакції | Проблеми

Рахунки (розгорнути) (Редагувати)

Персональні дані (розгорнути) (Редагувати)

Записи (розгорнути) (Редагувати) (Додати)

**2015-04-04 Звернення для Наталія Іваненко** (Видалити) (Розгорнути все) (Згорнути Все)

Pregnancy CDSS Form by Administrator Administrator (Згорнути)

Діагностичні дані:  
Вагітність: **Перша**  
Вірогідний діагноз: **Передчасні роди**

Детальніше про вірогідні діагнози:

Діагноз	Кільк. значущих симптомів
Своєчасні роди	6
Передчасні роди	17

Анкету створено:  
- Користувач: admin  
- Дата і час: 2015-04-04 22:43:00

Анкету востаннє змінювали:  
- Користувач: admin  
- Дата і час: 2015-05-04 15:28:07

Рис. 4. Діагностичний звіт СППР діагностики патологій невиношування вагітності у формі представлення звернення пацієнта у MIC OpenEMR.

На даному етапі розробки в формі анкети передбачено використання двох типів питань – з вибором єдиного варіанту відповіді або кількох варіантів одночасно (1 та 2 на рис. 3 відповідно).

Процес керування формою анкети та форми-звітами, в рамках застосування MVC-підходу, виконано у вигляді контролера PatientExam\_Form\_Controller, що реалізує такі методи:

- new\_action() та view\_action (\$form\_idexam) – забезпечує створення нової та відображення існуючої діалогової форми анкетування пацієнта (рис. 3);

- report\_action (\$form\_idexam) – забезпечує відображення даних СППР у формі звертання пацієнта у MIC OpenEMR (рис. 4) за допомогою класу представлення SymptByPatient\_Form2Report.class.php;

- print\_action (\$form\_idexam) – забезпечує відображення протоколу анкетування пацієнта та діагностичного висновку СППР у MIC OpenEMR (рис. 5) за допомогою класу представлення SymptByPatient\_Form2Print.class.php.

#### 2.4. Реалізація алгоритму процесу прийняття рішення

Прийняття діагностичних рішень у даній СППР здійснюється за допомогою таких алгоритмів:

- вірогіднісний алгоритм на основі статистичних даних роботи [25], реалізований А. В. Семенцем;
- алгоритм прийняття рішення на основі побудови «дерева рішень», описаний в роботах [15, 17, 18], реалізований В. П. Марценюком.

Вірогіднісний алгоритм реалізовується методом probability\_deceases\_action(\$symptom\_array) контролера PatientExam\_Form\_Controller (лістинг 2).

#### Лістинг 2. Метод, що реалізує вірогіднісний алгоритм прийняття рішення

```
public function probability_deceases_action($symptom_array) {
    //prepare default deceases array
    $curr_decease_multi = array();
    $decease = new Deceases2_Model();
    $deceases_arr = $decease->Find('');
    foreach ($deceases_arr as $dec) {
```

Pregnancy CDSS Form

<b>Пацієнт:</b> Код: 2 Прізвище: <b>Іваненко</b> П'ять: <b>Наталія</b> По-батькові: <b>Миколайівна</b> Дата народження: <b>04/16/1994</b> Адреса: Телефон:	<b>Діагностичні дані:</b> Вагітність: <b>Повторна</b> Остаточний діагноз: <b>Передчасні роди</b>						
<b>Вірогідна (більш імовірна) оцінка:</b> Вірогідний діагноз: <b>Передчасні роди</b> Детальніше:							
<table border="1"> <thead> <tr> <th>Діагноз</th> <th>Кільк. зустрічних симптомів</th> </tr> </thead> <tbody> <tr> <td>Своєчасні роди</td> <td>6</td> </tr> <tr> <td>Передчасні роди</td> <td>21</td> </tr> </tbody> </table>		Діагноз	Кільк. зустрічних симптомів	Своєчасні роди	6	Передчасні роди	21
Діагноз	Кільк. зустрічних симптомів						
Своєчасні роди	6						
Передчасні роди	21						

Дані анкетування:

Параметр	Значення	Вірог.діагноз:
<b>1. Анкетні дані вагітної</b>		
1.2. Сезон року	2. Весна	Своєчасні роди
1.3. Адреса проживання	1. Місто	Своєчасні роди
1.4. Реєстрація шлюбу	1. Зареєстрований	Своєчасні роди
1.5. Вид роботи	3. Домогосп	Своєчасні роди
1.6. Швидкості роботи	2. Переозлоджений	Передчасні роди
1.7. Стабільність	5. Порушення менструації	Передчасні роди
1.8. Шкідливі звички	1. Кова	Передчасні роди
1.9. Фізична активність	2. Повільна	Передчасні роди
1.10. Взяття на облік в ЖК	1. До 12 тижнів	Своєчасні роди
<b>2. Стан при поступленні в стаціонар</b>		
2.2. Вага жінки	5. 81-90 кг.	Передчасні роди
2.3. Котра вагітність	4. Четверта	Передчасні роди
2.4. Котріполога	3. Треті	Передчасні роди
2.5. Розмір тазу	1. Нормальні	Своєчасні роди
<b>4. Анамнез вагітної</b>		
4.5. Менструації встановились	1. Одразу	Своєчасні роди
4.7. Регулярність менструацій	1. Так	Своєчасні роди
4.8. Тривалість менструацій	2. 3-4 дні	Передчасні роди

Рис. 5. Фрагмент протоколу анкетування пацієнта та діагностичний висновок СППР діагностики патологій невиношування вагітності для MIC OpenEMR.

```

//default - each decease probability
=1
$curr_decease_multi[$dec->id][py]=1;
$curr_decease_multi[$dec->id][pn]=1;
$curr_decease_multi[$dec->id]
[count]=0;
}
$expectdeceasecount = 0;
//process form submissions
$deceasesymptopt = new DeceasesSymptOpt2_
Model();

foreach ($symptom_array as $sympt_
id=>$sympt_options){
    foreach ($sympt_options as $key=>$id_
sympt_opt){
        if ($deceasesymptopt->Load('id_sympt_
opt='.$id_sympt_opt))
            {
                $curr_decease_multi[$deceasesymptopt-
>id_deceases][py]=$curr_decease_
multi[$deceasesymptopt->id_deceases]
[py]*$deceasesymptopt->py;
            }
        }
    }

```



```

$curr_decease_multi[$deceasesymptopt->id_deceases][pn]=$curr_decease_multi[$deceasesymptopt->id_deceases][pn]*$deceasesymptopt->pn;
$curr_decease_multi[$deceasesymptopt->id_deceases][count]=$curr_decease_multi[$deceasesymptopt->id_deceases][count]+1;
//define most expected decease
if ($curr_decease_multi[$deceasesymptopt->id_deceases][count] > $expectdeceasecount) {
    $this->expectdeceaseid = $deceasesymptopt->id_deceases;
    $expectdeceasecount = $curr_decease_multi[$deceasesymptopt->id_deceases][count];
}
}
}
//Get most expected decease name
$decease->Load('id='.$this->expectdeceaseid);
$this->expectdeceasename = $decease->dec_name;
//Get final decease name (if exist)
if ($this->id_finaldecease > 0) {
    $decease->Load('id='.$this->id_finaldecease);
    $this->finaldecease = $decease->dec_name;
} else {
    $this->finaldecease = '';
}
return $curr_decease_multi;
}

```

#### 2.4.1. Розробка GAE-сервісу Decision Tree

Математично задача індукції дерева рішень формулюється таким чином. Маємо множину  $D$ , що містить  $N$  наборів навчальних даних. При цьому кожен  $i$ -й набір  $(A_1^i, A_2^i, \dots, A_p^i, C^i)$  складається з вхідних даних – атрибутів  $A_1, \dots, A_p$  та вихідних даних – атрибуту класу  $C$ . Атрибути  $A_1, \dots, A_p$  можуть приймати як чисельні, так і категоріальні значення. Атрибут класу  $C$  приймає одне з  $K$  дискретних значень:  $C \in \{1, \dots, K\}$ . Метою є прогнозування деревом рішень значення атрибуту класу  $C$

на основі значень атрибутів  $A_1, \dots, A_p$ . При цьому слід максимізувати точність прогнозування атрибуту класу, а саме  $P\{C = c\}$  на термінальних вузлах для довільного  $c \in \{1, \dots, K\}$ . Алгоритми індукції дерев рішень автоматично розбивають на вузлах значення чисельних атрибутів  $A_i$  на два інтервали:  $A_i \leq x_i$  та  $A_i > x_i$ , а категоріальних атрибутів  $A_j$  – на дві підмножини:  $A_j \in S_j$ ,  $A_j \notin S_j$ . Розбиття чисельних атрибутів ґрунтується, як правило, на мірах на основі ентропії, або індекси Джині [28]. Процес розбиття рекурсивно повторюється до тих пір, поки не спостерігатиметься покращення точності прогнозування. Останній крок включає вилучення вузлів для уникнення оверфітінгу моделі. В результаті ми повинні отримати множину правил, що йдуть від кореня до кожного термінального вузла, містять нерівності для чисельних атрибутів та умови включення для категоріальних атрибутів.

Нашою метою роботи є розробити метод індукції дерева рішень з можливістю програмної реалізації у вигляді клінічної експертної системи.

**Метод індукції дерева рішень.** За основу взято таку рекурсивну процедуру роботи [28].

Генерація дерева рішень

**Вхідні дані:**  $D$  – множина навчальних наборів

даних  $(A_1^i, A_2^i, \dots, A_p^i, C^i)$ .

**Вихідні дані:** дерево рішень

**Метод:**

1. Створити вузол  $N$ .
2. Якщо усі набори в  $D$  належать до спільного класу  $C$ , тоді повернути вузол  $N$  як листок із назвою класу  $C$ .
3. Якщо список атрибутів (а отже і  $D$ ) є порожнім, тоді повернути вузол  $N$  як листок із назвою найпоширенішого класу в  $D$ .
4. Застосувати *Алгоритм відбору атрибуту* із списку атрибутів і для множини  $D$  з метою відшукування «найкращого» атрибуту поділу.
5. Вилучити атрибут поділу із списку атрибутів.
6. Для кожної умови поділу  $j$  для атрибуту поділу розглянемо  $D_j$  – множину наборів з  $D$ , що задовольняють умову поділу  $j$ .
7. Якщо  $D_j$  – порожня, тоді приєднати до вузла  $N$  листок під заголовком найпоширенішого класу в  $D_j$ , інакше – приєднати до  $N$  вузол, що повертається рекурсивним викликом методу *Генерація дерева рішень* з вхідними даними  $D_j$  та список атрибутів.

8. Кінець циклу кроку 6.
9. Повернути вузол  $N$ .

В основу *Алгоритму відбору атрибуту* на  $j$ -му кроці рекурсії покладено такий інформаційний показник:

$$Gain(A_i) = Info(D_j) - Info_{A_i}(D_j). \quad (1)$$

Тут

$$Info(D_j) = -\sum_{k=1}^K p_k^j \log_2(p_k^j) \quad (2)$$

– інформація, потрібна для класифікації набору  $(A_1, A_2, \dots, A_p)$  в  $D_j$ ,

$$Info_{A_i}(D_j) = \sum_{l=1}^{K_i} \frac{\#(D_j^l)}{\#(D_j)} Info(D_l) \quad (3)$$

– інформація, потрібна для класифікації  $(A_1, A_2, \dots, A_p)$  в  $D_j$  після поділу  $D_j$  на підмножини  $D_j^l$  відповідно до значень атрибуту  $A_i$ .

У формулі (2) ймовірність того, що довільний набір з  $D_j$  належить множині  $C_{k,D_j}$  оцінюється як

$p_k^j = \frac{\#(C_{k,D_j})}{\#(D_j)}$ , де  $C_{k,D_j}$  – множина наборів з  $D_j$ , для яких атрибут класу  $C=k$ . Тут  $\#(\bullet)$  – кількість елементів в множині.

У формулі (3)  $\frac{\#(D_j^l)}{\#(D_j)}$  – оцінка ймовірності того, що довільний набір з  $D_j$  належить множині  $D_j^l$ , де  $D_j^l$  – множина наборів з  $D_j$ , для яких атрибут  $A_i = a_i^l$ . Тут атрибут  $A_i \in \{a_i^1, a_i^2, \dots, a_i^{K_i}\}$ .

Отже,  $Gain(A_i)$  оцінює зменшення інформації, необхідної для класифікації довільного набору даних в  $D_j$  за рахунок відомого значення атрибуту  $A_i$ . Таким чином, з наявних атрибутів на кожному вузлі дерева рішень для умови поділу слід відбирати атрибут  $A_{i^*}$  з найбільшим значенням  $Gain(A_{i^*})$ . В результаті такого вибору для завершення процесу класифікації набору даних в  $D_j$  вимагатиметься найменше інформації.

Метод реалізовано в якості GAE-сервісу на мові програмування Java. Базу навчальних даних розгорнуто у Google Datastore, що є насправді високомасштабованою базою даних, що не ґрунтується на SQL. Для взаємодії з Datastore використовуються інструменти для побудови запитів у фреймворку Objectify. На рисунку (рис. 6)

представлено концептуальну модель GAE-сервісу. У класі DecisionTree безпосередньо реалізовано метод індукції дерева рішень. У клас DataManager надходять виклики від DecisionTree на виконання запитів до бази даних Datastore щодо отримання навчальних даних.

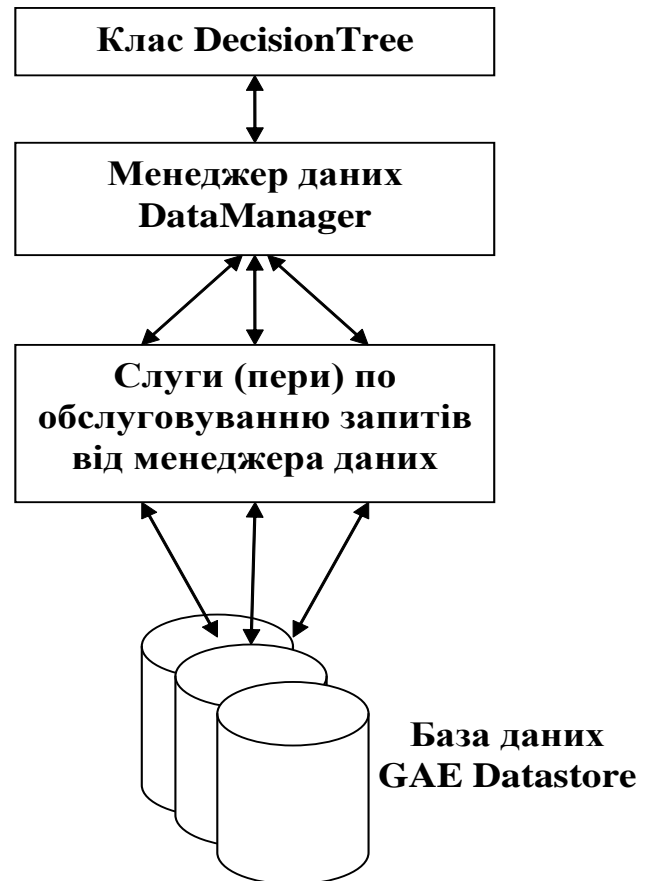


Рис. 6. Концептуальна модель GAE-сервісу індукції дерева рішень.

На рисунку (рис. 7) представлено діаграму класів для сховища Datastore. База даних складається з сутностей класів – Problem, призначеного для зберігання інформації про конкретну задачу (наприклад, прогнозування перебігу пологів); Attribute – для збереження описової інформації про атрибути навчальних наборів; Tuple – для зберігання навчального набору; CategorizedData – для значень атрибутів, включених в навчальний набір; ClassAttributeValue – для значення атрибуту класу в навчальному наборі. Оголошення класів-сутностей Datastore наведено нижче (лістинг 3):

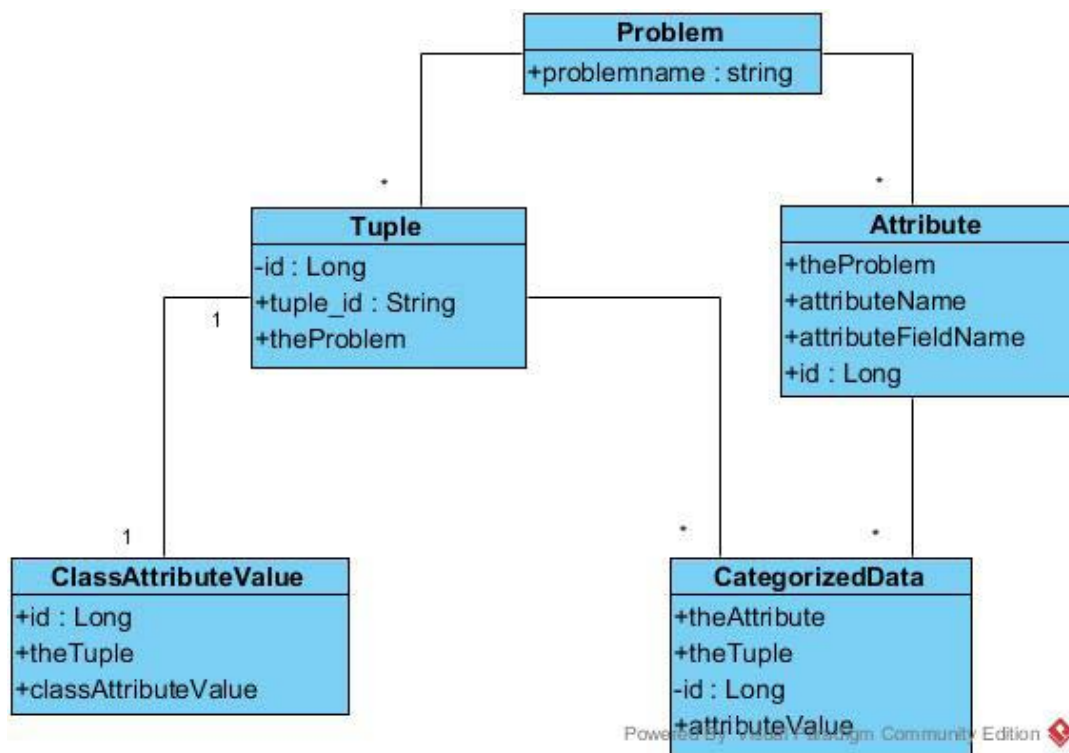


Рис. 7. Діаграма класів для зберігання навчальних наборів в GAE Datastore.

### Лістинг 3. Оголошення класів-сутностей Datastore GAE-сервісу

```

@Entity
public class Problem {
    @Id public String problemname;
}
@Entity
public class Attribute {
    @Parent com.googlecode.objectify.
    Ref<Problem> theProblem;
    @Id public Long id;
    public String attributeName;
    @Index public String
    attributeFieldName;
    @Index public Date date;
}
@Entity
public class Tuple {
    @Parent Ref<Problem> theProblem;
    @Index public String tuple_id;
    @Id public Long id;
}
@Entity
public class CategorizedData {
    @Load public Ref<Attribute>
    theAttribute;
    @Id Long id;

```

```

    @Parent public Key<Tuple> theTuple;
    public String attributeValue;
}
@Entity
public class ClassAttributeValue {
    @Id public Long id;
    @Parent public Key<Tuple> theTuple;
    public String classAttributeValue;
}

```

Програмні класи проекту включено до пакету com.google.gwt.decision\_tree.model. Сюди входять beans-класи Attribute, Attribute\_for\_list та Categorized Data для роботи з даними відповідних таблиць. Запити на основі фреймворка Objectify щодо отримання відповідних даних, включаючи розрахунки інформаційних показників, реалізовано в класі AttributeListPeer.

Клас DecisionTree має два елементи класу: m\_dataManager – менеджер даних та m\_htAttribute\_list – хеш-таблиця із списком атрибутів. Хеш-таблиця із списком атрибутів (у методах класу DecisionTree виступає під назвою htAttribute\_list) створюється для кожного вузла дерева рішень. Вона має два призначення – поряд із списком включених для даного вузла атрибутів зберігати умови поділу (splittingconditions), які перейшли до даного вузла

від вузлів-батьків. Кожен вузол дерева рішень є об'єктом класу `TextInBoxWithUserObject`. В якості об'єкта кожен вузол зберігає об'єкт класу `NodeObject`, декларація якого наведена нижче:

```
class NodeObject {
    Attribute attribute;
    Hashtable htAttribute_list;
    String splitting_criterion;
    String sLabel;
    public String toString() {
        if (splitting_criterion.
matches(«»)) { return sLabel; }
        else return «if `» + splitting_
criterion + «' then `» + sLabel + «'»;
    }
}
```

Тут `attribute` – атрибут, який повертається методом `Attribute_selection_method`, `splitting_criterion` – умова поділу, яка переходить від батьківського вузла, `sLabel` – надпис на вузлі. Хеш-таблиця `htAttribute_list` використовується для побудови наборів навчальних даних  $D_j$  для кожного із вузлів і має таку структуру:

Тип ключа	int
Тип об'єкта	Attribute_for_list
Структура об'єкта	Attribute attribute; Hashtable htSplitting_outcomes; String splitting_criterion; boolean included;

Тут `included` – булева змінна-прапорець належності атрибуту `attribute` до списку атрибутів даного вузла. Можна показати, що коли `included=true`, то вузол з назвою `attribute` є для даного вузла дочірнім (на певному нижчому рівні ієрархії). У випадку, коли атрибут `attribute` не входить до списку атрибутів для даного вузла (`included=false`), то вузол з назвою `attribute` є батьківським (на певному рівні ієрархії), а в змінній `splitting_criterion` зберігається умова поділу, якій підлягає даний вузол відносно батьківського вузла `attribute`.

Хеш-таблиця `htSplitting_outcomes` містить усі можливі наслідки (умови поділу) щодо атрибуту `attribute`.

Метод `Generate_decision_tree` є безпосередньою реалізацією методу індукції дерева рішень. Заголовок методу має вигляд:

```
private void Generate_decision_
tree(Hashtable htAttribute_list,
TextInBoxWithUserObject tbSubroot,
```

```
String splitting_criterion, String
tree_type)
```

В якості аргументів метод використовує кореневий вузол дерева, список пов'язаних з ним атрибутів `htAttribute_list` та умову поділу `splitting_criterion`. Шляхом рекурсивного виклику методу `Generate_decision_tree` будується дерево рішень.

З метою візуалізації представлення дерева використано пакет `org.abego.treelayout`. Розв'язано проблему використання даного пакету на платформі GAE. При цьому дерево рішень візуалізується на сторінці на основі SVG-формату, що підтримується більшістю сучасних веб-браузерів.

Взаємодію СППР з сервісом `GAE Decision-Tree` організовано з використанням додаткової бібліотеки `yii2-curl` (<http://www.yiiframework.com/extension/yii2-curl/>), що забезпечує формування `HttpRequest` запитів та їх обробку. Наступний метод (лістинг 4) здійснює вибірку даних поточної анкети та передачу її на сервер, що забезпечує виконання сервісу `GAE Decision Tree` для обробки та отримання діагностичного висновку.

#### Лістинг 4. Метод, що реалізує взаємодію з сервісом `GAE Decision Tree`

```
public function decisiontreegae_
action($form_idexam) {
    //show form report on the
encounter page
    $form_idexam = intval($form_
idexam);
    //fetch form data
    $form_data = formFetch($this-
>table_name, $form_idexam);
    //construct row
    $submitArray = array();
    $row = array();
    $client_description =
array();
    $client_decease = array();
    $client_data = array();
    //prepare array with form
data
    $client_description = array_
merge($client_description, ['url'=>$_
SERVER['SERVER_NAME'], 'form_name' =>
$this->form_name, 'exam_id'=>$form_
idexam, 'patient_id'=>$this->form_
pid]);
    $row = array_merge($row,['client_
```

```
description'=>$client_description]);
    $row = array_merge($row,['client_
decease'=>$client_decease]);
    //var_dump($client_
description);
    //prepare array with symptoms
list
    $Symptoms = Symptoms_
Model::all();
    //process all symptoms:
    foreach ($Symptoms as
$key=>$Symptom) {
        $symptoptbyperson = new
SymptByPatient2_Model();//prepare tmp
record
        //check symptom type:
        if (Symptoms_Model::is_
multy($Symptom->id)) {
            //Symptom can have
multiple options
            foreach ($Symptom-
>symptoptions as $optkey=>$SympOption)
{
                //load EACH symptom
options by this patient
                $symptoptbyperson-
>Load('(id_exam='.$form_idexam.')
AND(pid='.$this->form_pid.)AND(id_
symptom='.$Symptom->id.)AND(id_symp_
opt='.$SympOption->id.)');
            }
        } else {
            //load single symptom
option by this patient
            $symptoptbyperson-
>Load('(id_exam='.$form_idexam.')
AND(pid='.$this->form_pid.)AND(id_
symptom='.$Symptom->id.)');
            $tmpsymptoptdata =
new SympOptions2_Model();//prepare tmp
record
            $tmpsymptoptdata-
>Load('id='.$symptoptbyperson->id_
symp_opt);
            $client_data =
array_merge($client_data,[$Symptom-
>id=>['symp_id'=>$Symptom->id,'symp_
name'=>$Symptom->symp_name,'opt_
id'=>$symptoptbyperson->id_symp_opt,
'opt_name'=>$tmpsymptoptdata->opt_
name]]);
        }
    }
}
```

```
}}
    $row = array_merge($row,
['client_data'=>$client_data]);
    //ad new row to array
    $submitArray[] = $row;
    //convert to json
    $submitArrayjson = json_
encode($submitArray);
    var_dump($submitArray);
    //TODO: submit array to
GAE....
    $url = http://decisiontree-1013.
appspot.com/patientdata';
    // The submitted form data, encoded as
query-string-style
    $c = curl_init($url);
    curl_setopt($c, CURLOPT_POST, true);
    curl_setopt($c, CURLOPT_POSTFIELDS,
$submitArrayjson);
    curl_setopt($c, CURLOPT_RETURNTRANS-
FER, true);
    $page = curl_exec($c);
    curl_close($c);
    return;
}
```

## 2.5. Реалізація засобів керування платформою СППР

При розробці даної СППР авторами передбачено можливість її використання для вирішення різноманітних діагностичних задач, що допускають застосування алгоритмів прийняття рішення, вказаних в п. 2.2. Для цього розроблено спеціальний засіб керування платформою СППР у вигляді окремого адміністративного веб-модуля, також інтегрованого з МІС OpenEMR (рис. 8).

Вказаний адміністративного веб-сайту СППР містить засоби для:

- зміни переліку можливих діагностичних висновків;
- редагування всіх складових елементів анкети пацієнта;
- модифікації статистичних параметрів взаємозв'язків між діагностичними ознаками та висновками.

При розробці адміністративного веб-модуля СППР застосовано сучасний РНР-фреймворк Yii 2 (<http://www.yiiframework.com/>). Для прискорення процесу розробки авторами застосовано засоби автоматичної генерації коду, що наявні в Yii 2 – Gii (<https://github.com/yiisoft/yii2-gii>). Даний модуль

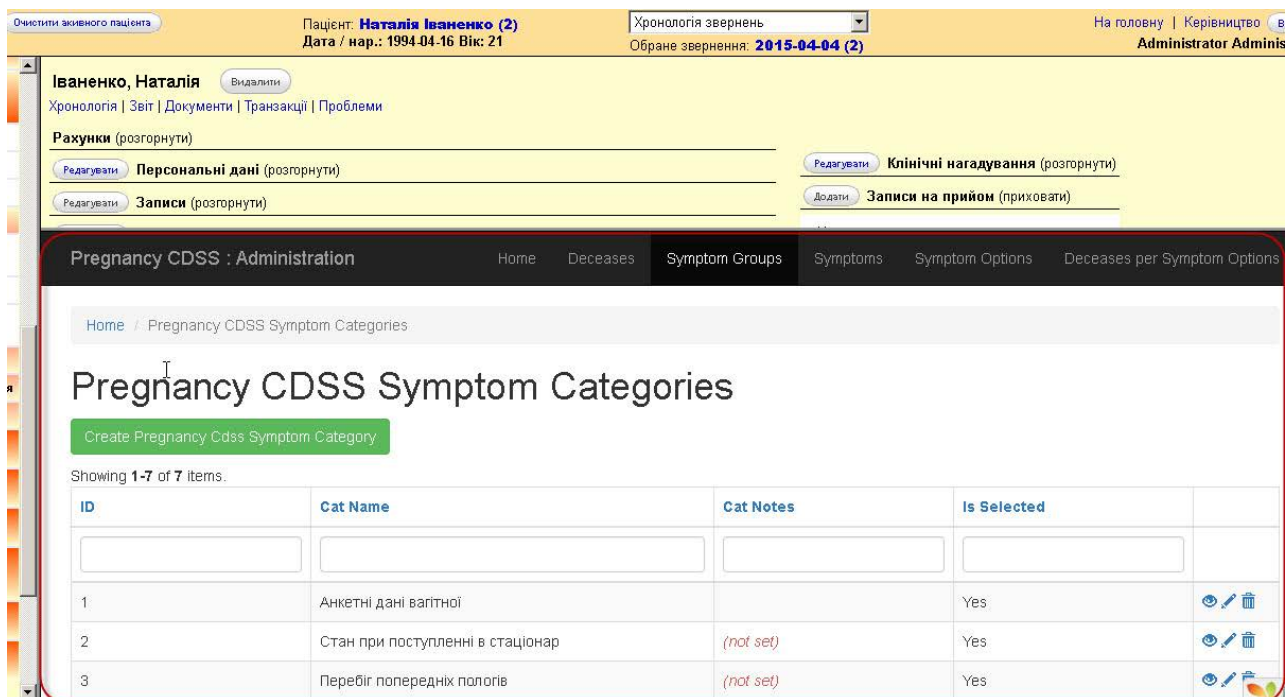


Рис. 8. Інтегрований засіб керування платформою СППР діагностики патологій невиношування вагітності у MIC OpenEMR.

дозволяє суттєво заощадити час, особливо при реалізації типових CRUD-операцій (Create/Read/Update/Delete) для таблиць БД додатку. Згенерований Gii код відповідає стандартам розробки об'єктно-орієнтованих додатків на мові PHP, завдяки чому розробник може легко добавляти необхідний функціонал.

Засобами Gii авторами було повністю реалізовано код CRUD-операцій для таблиць БД СППР.

Типовий приклад веб-форми, отриманої засобами Gii, показано на рисунку (рис. 9), де лише меню (1 на рис. 9) формується розробником самостійно. Інші елементи, зокрема: кнопка створення нового запису (2 на рис. 9), елементи сортування та фільтрації таблиці даних (3 та 4 на рис. 9), кнопки керування окремим записом в таблиці (5 на рис. 9) згенеровані автоматично і повністю функціональні.

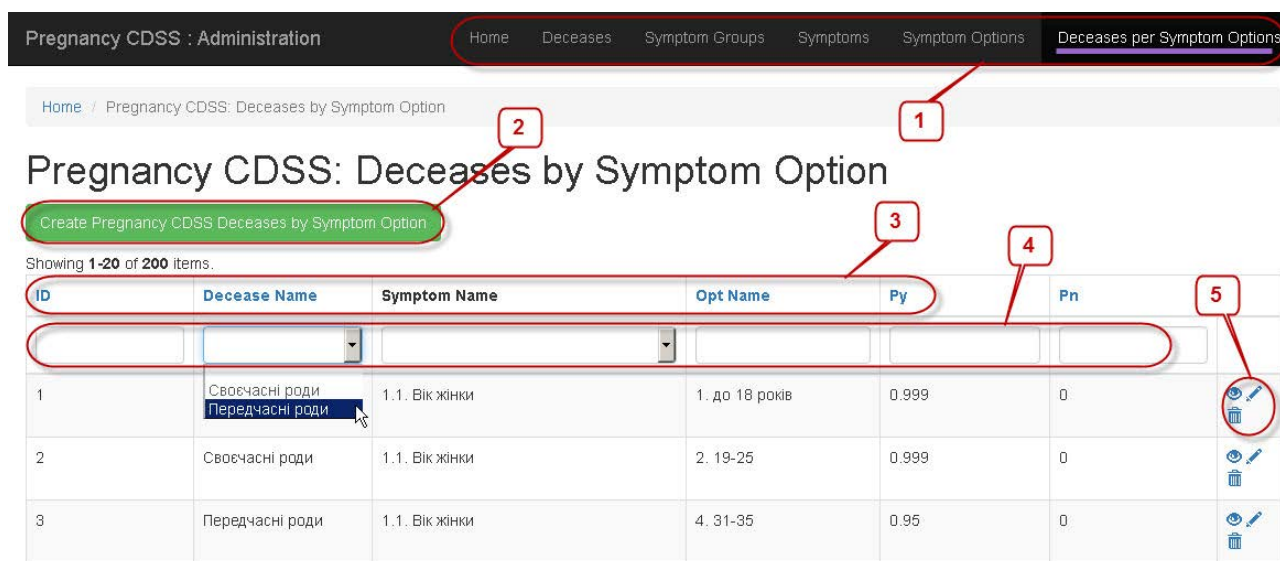


Рис. 9. Елементи типової форми веб-сайту керування платформою СППР, згенеровані засобами автоматичної генерації коду Gii.

Слід зазначити, що вищесказане стосується лише випадку, коли код генерується для однієї таблиці БД. У випадку наявності зв'язків між таблицями БД, елементи сортування та фільтрації таблиці даних (3 та 4 на рис. 9), а, також, форми редагування та перегляду даних потребують модифікації розробником для коректного відображення зв'язаних полів, наприклад, як описано в документації (<http://www.yiiframework.com/doc-2.0/guide-output-data-widgets.html>). Приклад модифікованої форми (для редагування переліку симптомів) показано на рисунку (рис. 10), де поля вводу (елемент вводу інформації по-замовчуванню) було замінено на:

– список, що розкривається – для вибору категорії симптому (поле «Id Category»);

– поля – «прапорці» для керування типом поля симптому та його видимістю (поля «Is Multi» та «Is Selected»).

Взаємодія веб-модуля адміністрування СППР з сервісом GAE Decision Tree організовано також з використанням вищевказаної додаткової бібліотеки yii2-curl що забезпечує формування HTTPRequest запитів та їх обробку. Наступний метод (лістинг 5) здійснює вибірку даних всіх анкет з встановленим остаточним діагнозом та передачу їх на сервіс GAE Decision Tree для використання в якості «навчальних» наборів даних.

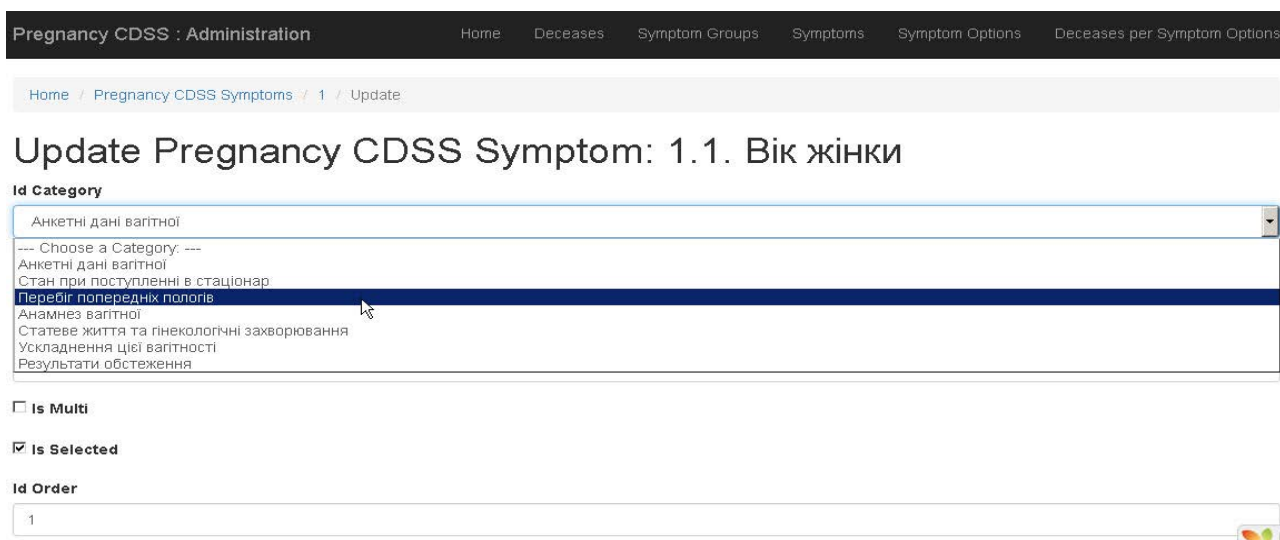


Рис. 10. Елементи форми веб-сайту керування платформою СППР, після модифікації розробником.

### Лістинг 5. Метод, що реалізує взаємодію з сервісом GAE Decision Tree

```
public function actionDecisiontreeeducationsubmit() {
    $searchModel = new PregnancyCdssPatientExamSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
    $submitReport = 'Not implemented';
    $formsdataArray = $dataProvider->getModels();

    $symptomssearchModel = new PregnancyCdssSymptomsSearch();
    $symptomsdataArray = $symptomssearchModel->find()-
```

```
>where(['is_selected'=>'1'])->all();
    $submitArray = array();
    foreach ($formsdataArray as $formsDataObj) {
        if ($formsDataObj->id_finaldecease) {
            //construct row
            $row = array();
            $client_description = array();
            $client_decease = array();
            $client_data = array();
            $client_description = array_merge($client_description, ['url'=>$_SERVER['SERVER_NAME'], 'form_name' => 'Pregnancy CDSS Form', 'exam_id'=>intval($formsDataObj->id),
```

```

'patient_id'=>intval($formsDataObj->pid));
        $client_decease =
array_merge($client_decease, ['decease_
id'=>$formsDataObj->id_finaldecease,
'decease_name'=>$formsDataObj->finaldecease]);
        $row = array_merge($row,['client_
description'=>$client_description]);
        $row = array_merge($row,['client_
decease'=>$client_decease]);
        foreach ($symptomsDataArray
as $symptomsDataObj) {
            //submit ALL symptoms
record
            if ($symptomsDataObj->is_multi==1) {
                //multiple choice
symptom
                $symptOptSearchModel
= new PregnancyCdssSymptOptionsSearch(
);
                $symptOptDataArray
= $symptOptSearchModel->find()->where(['id_symptom'=>$symptomsDataObj->id])->all();
                foreach
($symptOptDataArray as $symptOptDataObj)
{
                    $patientChoice
= PregnancyCdssSymptoptByPatient::findOne(['id_exam'=>$formsDataObj->id, 'pid'=>$formsDataObj->pid, 'id_symptom'=>$symptomsDataObj->id, 'id_sympt_opt'=>$symptOptDataObj->id]);
                    if ($patientChoice)
{
                        $patientChoiceName
= PregnancyCdssSymptOptions::findOne(['id'=>$patientChoice->id_sympt_opt]);
                        $client_data = array_
merge($client_data,[$symptomsDataObj->id=>['symp_id'=>$symptomsDataObj->id, 'symp_name'=>$symptomsDataObj->symp_name, 'opt_id'=>$patientChoice->id_sympt_opt, 'opt_name'=>$patientChoiceName->opt_name]]);
                    }
                } else {
                    //single choice
symptom
                    $patientChoice
= PregnancyCdssSymptoptByPatient::findOne(['id_exam'=>$formsDataObj->id, 'pid'=>$formsDataObj->pid, 'id_symptom'=>$symptomsDataObj->id]);
                    $patientChoiceName
= PregnancyCdssSymptOptions::findOne(['id'=>$patientChoice->id_sympt_opt]);
                    $client_data = array_
merge($client_data,[$symptomsDataObj->id=>['symp_id'=>$symptomsDataObj->id, 'symp_name'=>$symptomsDataObj->symp_name, 'opt_id'=>$patientChoice->id_sympt_opt, 'opt_name'=>$patientChoiceName->opt_name]]);
                }
            }
            $row = array_merge($row,
['client_data'=>$client_data]);
            //ad new row to array
            $submitArray[] = $row;
            // $submitArray[] = ['exam_
id'=>intval($formsDataObj->id),
'patient_id'=>intval($formsDataObj->pid),
'decease'=>$formsDataObj->id_finaldecease];
        } }
        if ($submitArray) {
            //convert to json
            $submitArrayjson =
Json::encode($submitArray);
            $subarrsize =
sizeof($submitArray);
            //Init curl
            $curl = new curl\Curl();
            //post http://contactmgr.loc/
            $submitReport = $curl->setOption(
                CURLOPT_POSTFIELDS,
                http_build_
query(array(
                    'submitArrayjson'
=> $submitArrayjson
                ))
            )->post('http://contactmgr.
loc/site/yii2curltest');
        }
    }
}

```



```

return $this->render( 'dectreesubmit',
[
    'searchModel' =>
$searchModel,
    'dataProvider' =>
$dataProvider,
    'submitReport' =>
$submitReport,
    'submitArray' =>
$submitArray,
    'submitArrayjson' =>
$submitArrayjson,
    'subarrsize' =>
$subarrsize,
]);
}

```

Процес підключення розробленої СППР до існуючого серверу МІС OpenEMR здійснюється

стандартними для даної МІС методами ([http://www.open-emr.org/wiki/index.php/OpenEMR\\_Contributed\\_Forms](http://www.open-emr.org/wiki/index.php/OpenEMR_Contributed_Forms)). Користувач повинен мати права адміністратора системи:

1. Скачати вихідний код СППР діагностики патологій невиношування вагітності з репозиторію на сервері GitHub ([https://github.com/semteacher/pregnacy\\_cdss](https://github.com/semteacher/pregnacy_cdss)).

2. Скачати вихідний код адміністративного веб-модуля СППР діагностики патологій невиношування вагітності з репозиторію на сервері GitHub().

3. Скопіювати вихідний код СППР та адміністративного веб-модуля у нові папки на сервері МІС OpenEMR (наприклад, {openemr\_root}\interface\forms\pregnacy\_cdss\ та {openemr\_root}\interface\forms\pregnacy\_cdss\admin\ відповідно).

4. Провести ініціалізацію модуля (плагіна) СППР діагностики патологій невиношування вагітності штатними засобами МІС OpenEMR (рис. 11).

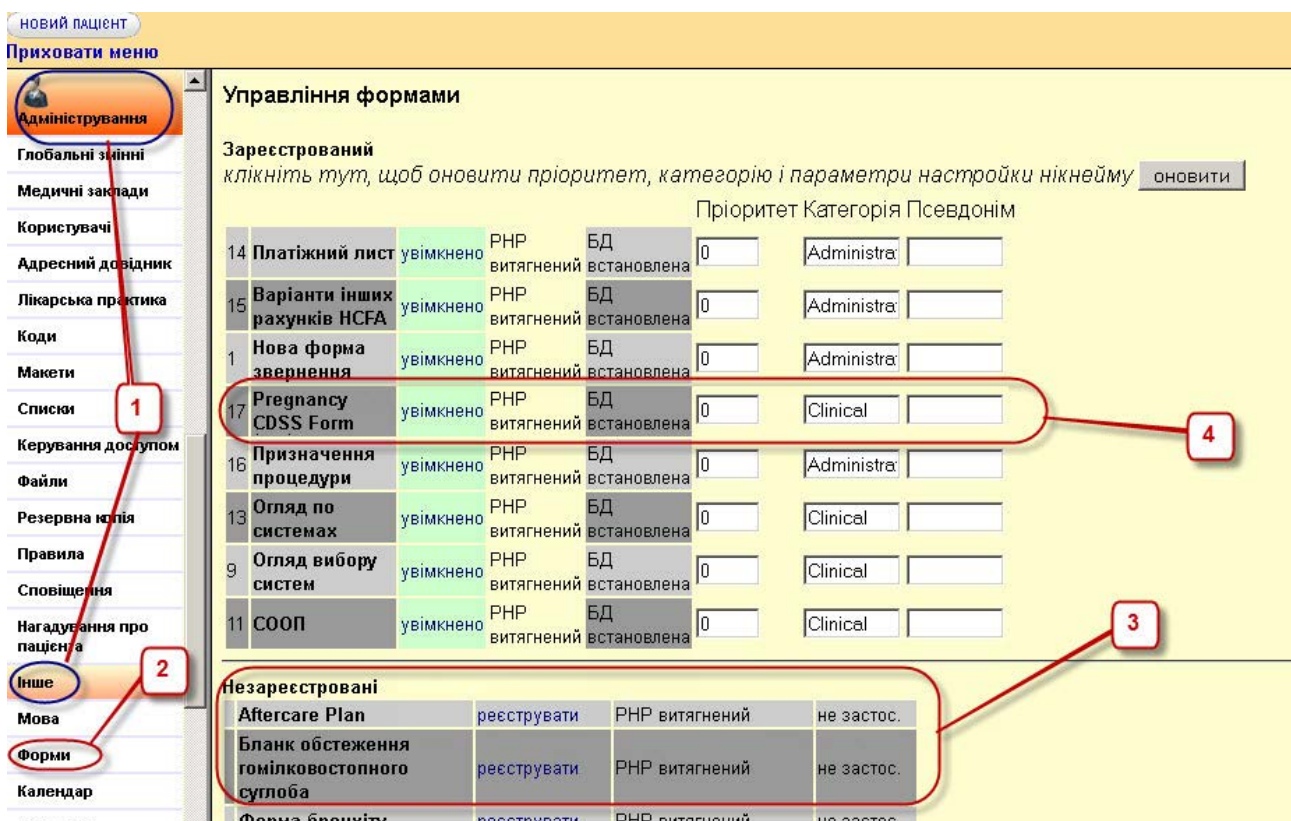


Рис. 11. Послідовність встановлення модуля (плагіна) СППР діагностики патологій невиношування вагітності на сервер МІС OpenEMR.

Після виконання вказаних кроків модуль (плагін) СППР діагностики патологій невиношування ва-

гітності буде доступним у розділі «Clinical» форми звернення (Encounter) пацієнта (рис. 12).

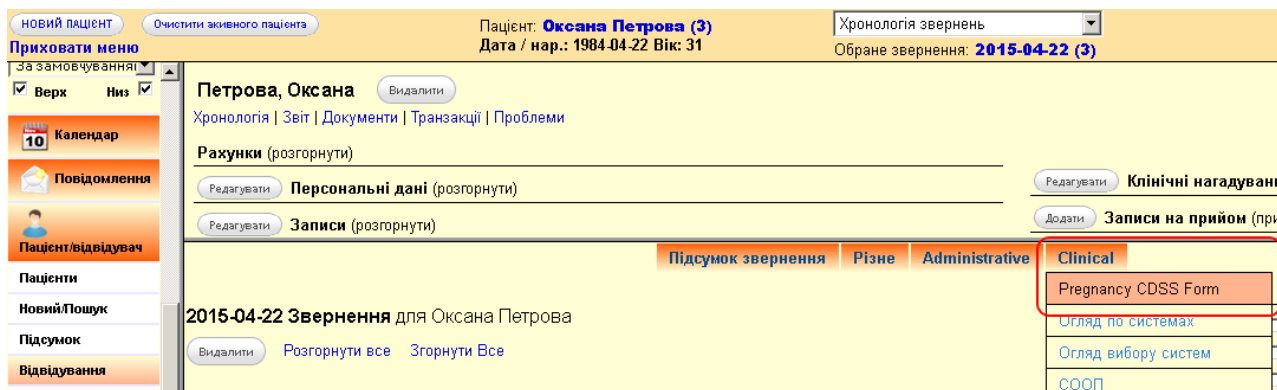


Рис. 12. Модуль (плагін) СППР діагностики патологій невиношування вагітності у формі звернення (Encounter) пацієнта MIC OpenEMR.

**Висновки.** Застосування в щоденній практичній діяльності сучасного лікаря клінічних СППР є необхідною умовою підвищення якості надання медичної допомоги. Перспективним застосуванням СППР є, зокрема, діагностика патологій невиношування вагітності та запобігання передчасним пологам.

Науковцями ТДМУ досліджуються фактори, що впливають на патологічний перебіг і спричиняють невиношування вагітності. Базуючись на їх результатах у 2013 р. А. В. Семенець, М. М. Жилияєв та С. М. Геряк розробили прототип СППР діагностики патологій невиношування вагітності, що реалізує модель вірогіднісного (статистичного) підходу в процесі прийняття діагностичного рішення.

Виходячи з досвіду дослідної експлуатації вказаного прототипу СППР в Тернопільському обласному перинатальному центрі «Мати і дитина» авторами розроблено спеціалізовану СППР у вигляді модуля (плагін) для вільно-розповсюджуваної MIC EMK OpenEMR. Авторами розроблено інформа-

ційну модель БД модуля (плагіна) та реалізовано діалоговий компонент СППР з використанням відкритих програмних API в MIC EMK OpenEMR.

Одночасне використання двох різних підходів до процесу прийняття діагностичних рішень – вірогіднісного та алгоритму прийняття рішення на основі побудови «дерева рішень» – у даній СППР дає можливість як підвищити якість діагностики, так і проаналізувати ефективність та надійність вказаних алгоритмів.

Завдяки використанню MVC-підходу до розробки СППР модуля (плагіна) та сучасних фреймворків для розробки окремого модуля адміністрування плагіну авторами отримано програмну платформу СППР для MIC EMK OpenEMR.

Перспективою подальших досліджень є адаптація розробленої платформи СППР для вирішення якнайширшого кола діагностичних задач, шляхом обробки інформації, що накопичується в MIC EMK OpenEMR.

## Література

1. Aminpour F. Utilization of open source electronic health record around the world: a systematic review / F. Aminpour, F. Sadoughi, M. Ahamdi // Journal of research in medical sciences : the official journal of Isfahan University of Medical Sciences. – 2014. – Vol. 19, № 1. – P. 57–64.
2. Авраменко В. І. Формування основних напрямків розвитку інформаційних технологій в охороні здоров'я України на основі світових тенденцій / В. І. Авраменко, В. О. Качмар // Український журнал телемедицини та медичної телематики. – 2011. – Т. 9. – № 2. – С. 5–15.
3. Семенець А. В. Організаційно-методичні підходи впровадження EMR-систем в охороні здоров'я України / А. В. Семенець // Медична інформатика та інженерія. – 2013. – № 3. – С. 35–43.
4. Global healthcare it market analysis and segment forecasts to 2020 – healthcare it industry, outlook, size, application, product, share, growth prospects, key opportunities, dynamics, trends, analysis, healthcare it report – grand view research inc / [Електронний ресурс] – Режим доступу до документу: <http://www.grandviewresearch.com/industry-analysis/healthcare-it-market>.
5. Reynolds C. J. Open source, open standards, and health care information systems / C. J. Reynolds, J. C. Wyatt // Journal of medical Internet research. – 2011. – Vol. 13, № 1. – P. e24.
6. List of open-source healthcare software – Wikipedia, the free encyclopedia / [Електронний ресурс] – Режим доступу до документу :

[http://en.wikipedia.org/wiki/List\\_of\\_open-source\\_health-care\\_software#Electronic\\_health\\_or\\_medical\\_record](http://en.wikipedia.org/wiki/List_of_open-source_health-care_software#Electronic_health_or_medical_record) .

7. Fritz F. Success criteria for electronic medical record implementations in low-resource settings: a systematic review / F. Fritz, B. Tilahun, M. Dugas // *Journal of the American Medical Informatics Association*. – 2015. – Vol. 22, № 2. – P. 479–488.
8. Семенець А. В. Про досвід міграції даних пацієнтів при впровадженні вільно розповсюдженого програмного забезпечення МІС ЕМК з відкритим кодом / А. В. Семенець // *Медична інформатика та інженерія*. – 2014. – Vol. 25, № 1. – С. 28–37.
9. Roshanov P. S. Features of effective computerised clinical decision support systems: meta-regression of 162 randomised trials / P. S. Roshanov, N. Fernandes, J. M. Wilczynski [et al.] // *BMJ (Clinical research ed.)*. – 2013. – Vol. 346, № feb14\_1. – P. f657.
10. Bright T. J. Effect of clinical decision-support systems: a systematic review. / T. J. Bright, A. Wong, R. Dhurjati [et al.] // *Annals of internal medicine*. – 2012. – Vol. 157, № 1. – P. 29–43.
11. Jaspers M. W. M. Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings / M. W. M. Jaspers, M. Smeulers, H. Vermeulen, L. W. Peute // *Journal of the American Medical Informatics Association : JAMIA*. – 2011. – Vol. 18, № 3. – P. 327–34.
12. Adoption of clinical decision support systems in a developing country: antecedents and outcomes of physician's threat to perceived professional autonomy / P. Esmailzadeh, M. Sambasivan, N. Kumar, H. Nezakati // *International Journal of Medical Informatics*. – 2015.
13. Integrating pharmacogenetic information and clinical decision support into the electronic health record / B. R. Goldspiel, W. A. Flegel, G. DiPatrizio [et al.] // *Journal of the American Medical Informatics Association : JAMIA*. – 2014. – Vol. 21, № 3. – P. 522–528.
14. Марценюк В. П. Медична інформатика. інструментальні та експертні системи / В. П. Марценюк, А. В. Семенець // *Тернопіль : Укрмедкнига*, – 2004.
15. Борис Р. М. Алгоритм класифікації політравм методом індукції дерева рішень / Р. М. Борис, В. П. Марценюк // *Медична інформатика та інженерія*. – 2013. – № 2. – С. 12–17.
16. Марценюк В. П. Розробка клінічної експертної системи, що ґрунтується на правилах, методом послідовного покриття / В. П. Марценюк, І. Є. Андрущак // *Наукові праці Чорноморського державного університету імені Петра Могили*. – 2014 – Т. 225, № 237. – С. 5–10.
17. Марценюк В. П. Про клінічну систему, що ґрунтується на правилах, на основі технології DataMining / В. П. Марценюк, О. О. Стаханська // *Медична інформатика та інженерія*. – 2014. – № 1. – С. 24–27.
18. Martsenyuk V.P. Qualitative Analysis of the Antineoplastic Immunity System on the Basis of a Decision Tree / V.P.Martsenyuk, I. Y.Andrushchak, I. S. Gvozdzetska // *Cybernetics and Systems Analysis*–2015. – Т. 51, № 3. – С. 461–470. DOI: 10.1007/s10559-015-9737-6
19. Hasmik Martirosyan. A decision-support system for expecting mothers and obstetricians / E. B. Hasmik Martirosyan, Monique Frize, Daphne E. Ong, Jeff Gilchrist // 6<sup>th</sup> European Conference of the International Federation for Medical and Biological Engineering MBEC 2014, 7–11 September 2014, Dubrovnik, Croatia. Springer International Publishing. – 2014. – Vol. 45. – P. 703–706.
20. Evaluation of a novel electronic genetic screening and clinical decision support tool in prenatal clinical settings / E. A. Edelman, B. K. Lin, T. Doksum [et al.] // *Maternal and child health journal*. – 2014. – Vol. 18, № 5. – P. 233–245.
21. Role of openehr as an open source solution for the regional modelling of patient data in obstetrics / C. Pahl, M. Zare, N. Mehrbakhsh [et al.] // *Journal of Biomedical Informatics*. – 2015.
22. Внутрішньоутробні інфекції: паралелі пренатального УЗД і патоморфологічного дослідження / Н. А. Васильєва, М. М. Жилиєв, А. В. Гвоздецький, М. М. Орел // *Інфекційні хвороби*. – 2014. – № 3. – P. 68–71.
23. Геряк С. М. Особливості продукції інтерлейкінів при своєчасних та ускладнених (антенатальна загибель плода) передчасних пологах / С. М. Геряк, М. М. Жилиєв // *Вісник наукових досліджень*. – 2013. – № 1. – С. 69–70.
24. Васильєва Н. А. Вплив torch-інфекцій на перебіг вагітності, передчасних, своєчасних пологів та перинатальну смертність плодів та дітей / Н. А. Васильєва, С. М. Геряк, М. М. Жилиєв // *Актуальні питання педіатрії, акушерства та гінекології*. – 2012. – № 2. – P. 60–62.
25. Жилиєв М. М. Порівняльна оцінка медико-статистичних показників перед своєчасними та передчасними пологами / М. М. Жилиєв // *Галицький лікарський вісник*. – 2012. – № 4, Т. 19. – С. 129–133.
26. Свідцтво про реєстрацію авторського права на твір № 51256 Комп'ютерна програма «Інформаційна система підтримки прийняття рішення »Вагітність» / А. В. Семенець, С. М. Геряк, М. М. Жилиєв від 16.09.2013.
27. Жилиєв Н. Н. Эффективность применения компьютерной скрининговой системы для диагностики и прогнозирования преждевременных родов / Н. Н. Жилиєв, С. Н. Геряк // *Вятский медицинский вестник*. – 2014. – №. 2. – С. 18–22.
28. Han J. and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 1<sup>st</sup> edition, 2001.