Практичне завдання № 2. Визначення функціональних / нефункціональних вимог і планування розробки програмного продукту

Виконала ст. гр. АС-123 Якимчук Катерина

Git-hub репозиторій: <https://github.com/kyakimchuk/diplom.git>

Папка зі звітами з дисципліни: PP

2.1 Функціональні вимоги

2.1) формування тезаурусів

Змагання за правилами ACM – команда з 3 осіб протягом 5-ти годин повинна вирішити 8-12 задач, використовуючи мови програмування C++, Java, Pascal чи інші. Для цього в учасників є тільки один комп’ютер. По мірі розв’язування команди надсилають готові відповіді на перевірку. Отримані результати оцінює комп’ютер. Якщо алгоритм працює правильно – команді нараховуються бали. В іншому випадку нараховуються штрафні бали.

Перевірка розв’язків на сервері – розв’язок перевіряється шляхом запуску на наборі тестів, що є однаковим для всіх і зараховується в тому випадку, якщо він видає вірні відповіді на всі тести.

Складність задачі – складність задачі, наприклад на сайті acm.timus.ru, обчислюється за формулою, що залежить від кількості авторів, які здали цю задачу, і від часу існування задачі. Чим менше нових авторів здають задачу за одиницю часу, тим вище її складність.

Список учасників – список тих людей, що здають задачі на сайтах з архівами задач. Найчастіше це студенти, оскільки саме вони можуть брати участь у конкурсі змагання.

Тренер – людина, яка проводить заняття з учасниками, дає їм завдання розв’язувати якісь задачі і відправляти їх на перевірку, в залежності від теми, що вони пройшли.

Учасник – людина, що здає задачі на сайтах з архівами задач. Теоретично це учасник змагання.

Id на сайтах timus.ru, e-olymp.com – унікальний ідентифікатор учасника, знаючи який можна прослідкувати, які задачі з якої спроби та з якими характеристиками він здав. Id на сайті timus.ru дається учаснику при реєстрації та складається с 6-ти цифр та двох великих латинських літер, на сайті e-olymp – id будь-який з латинських літер, знаків підкреслювання та цифр, його придумує сам учасник.

2.2) Структура декомпозиції робіт

|  |
| --- |
| **Організаційні процеси** |
| ***Навчання*** |
| Вивчення існуючої реалізації |
| Ввивчення архітектурного шаблону MVC |
| ***Управління проектом*** |
| Визначення цілей проекта та схем їх досягнення |
| Визначення і застосування необхідних коригувальних впливів |
| **Основні процеси** |
| ***Розробка*** |
| ВВ: Регістрація |
| ВВ: Авторизація |
| ВВ: Додавання учасника |
| ВВ: Редагування інформації про учасника |
| ВВ: Видалення учасника |
| ВВ: Вивести розв’язані задачі по темі |
| ***Супровід*** |
| Оптимізація ПО |
| Усунення дефектів ПО |
| **Допоміжні процеси** |
| ***Забезпечення якості*** |
| Формування характеристик якості |
| Підтримка характеристик |
| **Документування** |
| Керівництво користувача |
| Довідковий текст |

Визначення пріоритетів за принципом MoSCoW: усі вимоги мають пріоритет M (маємо це зробити).

Функціональний аналіз існуючих продуктів:

1) acm.timus.ru

2) e-olimp.com

3) Moodle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Продукти | Додати учасника | Редагувати учасника | Видалити участника | Вивести розв’язані задачі |
| 1 | - | - | - | + |
| 2 | - | - | - | + |
| 3 | + | + | + | - |

2.2 Нефункціональні вимоги

2.2.1) Зовнішні інтерфейси

Інтерфейс користувача має бути зручним, простим у використанні та інтуїтивно зрозумілим. Шрифт має бути усюди однаковим. Фон web-сторінок має бути світлим, має бути високий контраст між фоном та шрифтом. Меню має бути простим, назви кнопок та посилань повинні бути короткими і в той же час чітко відображати пов’язану з ними дію. Інформацію на сторінці, по можливості, слід виводити у формі таблиць для наочності.

Інтерфейс управління. З будь-якої сторінки web-програми доступне головне меню з навігацією по сайту, усіма доступними функціями даному користувачу. Гарячі клавіші залежать від браузеру, який буде встановлений у користувача. Усі повідомлення про успішність виконання операцій будуть виділятися зеленим фоном і кольором, про помилки – червоним.

Програмні інтерфейси: робота з web-програмою не залежить від виду операційної системи.

Обов’язково на комп’ютері має бути встановлений браузер.

Інтерфейси передачі даних: стандарт Ethernet, бездротовий інтерфейс Wifi.

2.2.2) Атрибути якості

Продуктивність:

-максимальний час відгуку програми на запити користувача - 3с.

-максимальна кількість одночасно обслуговуваних клієнтів – 100.

Надійність:

-мінімальний час безвідмовної роботи – 15 хвилин.

-відновлення даних після збоїв у системі.

2.3 Планування розробки

Діаграма концептуальних класів представлена на рисунку 1.

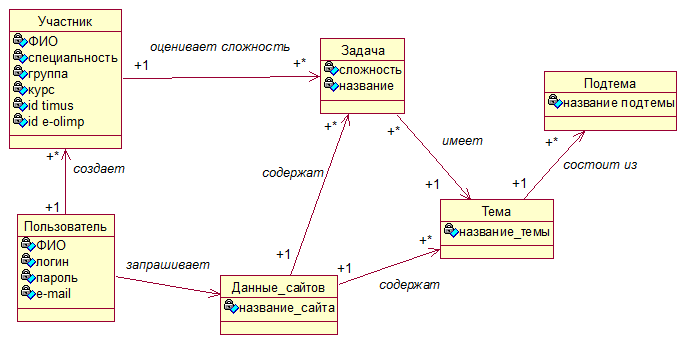


Рисунок 1 – Діаграма концептуальних класів

Технології розробки:

ОС: Windows 7

СУБД: MySQL

Програмні технології: Web

Структури даних: реляційна БД

Інструменти розробки:

IDE: JetBrains PhpStorm 9.0.2

VCS (система контролю версій): Git

Бібліотеки модульного тестування: PHPUnit

План розробки: Структура декомпозиції робіт представлена на рисунку 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название задачи | Длительность | Начало | Окончание | Предшественники |
| **Організаційні процеси** |  |  |  |  |
| ***Навчання*** |  |  |  |  |
| Вивчення існуючої реалізації | 2 дней | 02.04.16 | 03.04.16 |  |
| Ввивчення архітектурного шаблону MVC | 2 дней | 04.04.16 | 05.04.16 | 3 |
| ***Управління проектом*** |  |  |  |  |
| Визначення цілей проекта та схем їх досягнення | 2 дней | 06.04.16 | 07.04.16 | 4 |
| Визначення і застосування необхідних коригувальних впливів | 1 день | 08.04.16 | 08.04.16 | 12 |
| **Основні процеси** |  |  |  |  |
| ***Розробка*** |  |  |  |  |
| ВВ: Регістрація | 1 день | 09.04.16 | 09.04.16 | 20 |
| ВВ: Авторизація | 1 день | 10.04.16 | 10.04.16 | 21 |
| ВВ: Додавання учасника | 1 день | 11.04.16 | 11.04.16 | 11 |
| ВВ: Редагування інформації про учасника | 1 день | 12.04.16 | 12.04.16 | 7 |
| ВВ: Видалення учасника | 1 день | 13.04.16 | 13.04.16 | 10 |
| ВВ: Вивести розв’язані задачі по темі | 2 дней | 14.04.16 | 14.04.16 |  |
| ***Супровід*** |  |  |  |  |
| Оптимізація ПО | 1 день | 15.04.16 | 15.04.16 | 17 |
| Усунення дефектів ПО | 1 день | 16.04.16 | 16.04.16 | 24 |
| **Допоміжні процеси** |  |  |  |  |
| ***Забезпечення якості*** |  |  |  |  |
| Формування характеристик якості | 2 дней | 17.04.16 | 19.04.16 | 6 |
| Підтримка характеристик | 2 дней | 20.04.16 | 21.04.16 | 14 |
| **Документування** |  |  |  |  |
| Керівництво користувача | 1 день | 22.04.16 | 22.04.16 | 13 |
| Довідковий текст | 1 день | 23.04.16 | 23.04.16 | 23 |

Рисунок 2 – Структура декомпозиції робіт

Оцінка об’єму виконуваних робіт методом UCP:

Оцінка акторів

Таблиця 1 – Визначення типів акторів

|  |  |  |
| --- | --- | --- |
| **Назва актора** | **Тип актора** | **Вага** |
| Студент (учасник) | Складний | 3 |
| Тренер | Складний | 3 |
| Адміністратор | Складний | 3 |
| БД | Середній | 2 |

UAW = 3\*1 + 3\*1 + 3\*1 + 2\*1 = 11.

Оцінка варіантів використання

Таблиця 2 – Визначення типів варіантів використання за кількістю класів

|  |  |  |  |
| --- | --- | --- | --- |
| **Назва варіанту використання** | **Тип варіанту використання** | **Кількість класів** | **Вага** |
| Реєстрація | Простий | До 5 | 5 |
| Авторизація | Простий | До 5 | 5 |
| Додавання учасника | Простий | До 5 | 5 |
| Редагування інформації про учасника | Простий | До 5 | 5 |
| Видалення учасника | Простий | До 5 | 5 |
| Вивести розв’язані задачі по темі | Середній | Від 5 до 10 | 10 |

UUCW = 5\*5 + 10 = 35;

UCP = UAW + UUCW = 11+35=46.

Оцінка технічних факторів

Таблиця 3 – Технічні фактори

| **Фактор** | **Опис** | **Вага** | **Пояснення** | **Оцінка** |
| --- | --- | --- | --- | --- |
| T1 | Розподіленість системи | 2 | Інформує про потребу системи в розподілених обчисленнях | 3 |
| T2 | Час відгуку | 1 | Визначає ефективність системи з точки зору часу відгуку, потоку робіт тощо | 5 |
| T3 | Ефективність кінцевого користувача | 1 | Визначає ефективність користувача з точки зору його (її) сприйняття | 5 |
| T4 | Складність обробки | 1 | Визначає, чи будуть застосовуватись складні алгоритми для обробки даних | 1 |
| T5 | Фокус на повторному використанні коду | 1 | Визначає, чи будуть елементи коду системи використовуватись знову | 2 |
| T6 | Простота інсталяції | 0,5 | Визначає метод інсталяції та простоту інсталяції для кінцевого користувача, чи буде потреба в спеціалісті для інсталяції системи | 1 |
| T7 | Простота використання | 0,5 | Визначає узгодженість інтерфейсу користувача з його потребами | 4 |
| T8 | Портативність | 2 | Визначає, чи має застосування працювати в різних середовищах | 3 |
| T9 | Простота змін | 1 | Визначає, чи будуватиметься система в такий спосіб, щоб спростити її модифікації в майбутньому | 3 |
| T10 | Паралельні обчислення | 1 | Інформує, чи будуть мати в системі місце паралельні обчислення | 1 |
| T11 | Засоби захисту | 1 | Визначає чи потребуватиме система спеціальні засоби захисту даних чи системи | 3 |
| T12 | Доступ до третьої сторони | 1 | Визначає ступень використання системи зовнішніми системами або акторами | 2 |
| T13 | Потреби в спеціальному навчанні | 1 | Визначає, чи потрібно організувати тренінги для користувачів | 1 |

TFactor = 2\*3+1\*5+1\*5+1\*1+1\*2+0,5\*1+0,5\*4+2\*3+1\*3+1\*1+1\*3+1\*2+1\*1=37,5;

TCF = 0,6 + (0,01 \* TFactor) = 0,6 + (0,01\*37,5) = 0,975.

Оцінка зовнішніх факторів

Таблиця 4 – Зовнішні фактори

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Фактор | Опис | Вага | Пояснення | Оцінка |
|  |  |  |  |  |
| F1 | Знайомство з | 1,5 | Визначає, чи знайома команда з | 4 |
|  | процесом |  | предметною областю та технічними |  |
|  | розробки |  | аспектами вирішення проблеми клієнта. |  |
|  |  |  | Особливу увагу слід приділити знанню |  |
|  |  |  | методології, в якій виконується проект, а |  |
|  |  |  | також знанню мов моделювання системи |  |
|  |  |  |  |  |
| F2 | Досвід подібних | 0,5 | Загальна уява про досвід команди в | 4 |
|  | проектів |  | розробці програмного забезпечення |  |
|  |  |  |  |  |
| F3 | Досвід об’єктно- | 1 | Досвід в проектуванні об’єктно- | 3 |
|  | орієнтованої |  | орієнтованих застосувань, а також в |  |
|  | розробки |  | підтримці засобів для розробки |  |
|  |  |  | інформаційних систем |  |
|  |  |  |  |  |
| F4 | Досвідченість | 0,5 | Здатності аналітика отримати вимоги від | 2 |
|  | провідного |  | клієнта та знання щодо задач, які буде |  |
|  | аналітика |  | вирішувати система |  |
|  |  |  |  |  |
| F5 | Мотивація | 1 | Здатність команди займатись призначеною | 1 |
|  |  |  | задачею |  |
|  |  |  |  |  |
| F6 | Стабільність | 2 | Визначає, чи не будуть вимоги часто | 1 |
|  | вимог |  | змінюватись |  |
|  |  |  |  |  |
| F7 | Часткова | -1 | Визначає, наскільки великою є частка | 2 |
|  | зайнятість |  | працівників часткової зайнятості |  |
|  | працівників |  |  |  |
|  |  |  |  |  |
| F8 | Складність мови | -1 | Визначає, наскільки складно вивчити мову | 3 |
|  | програмування |  | програмування: |  |
|  |  |  | 0 – легко, за тиждень можна опанувати; |  |
|  |  |  | 1 – щонайменше 2 тижні потрібно для |  |
|  |  |  | опанування мови; |  |
|  |  |  | 2 – щонайменше місяць потрібно для |  |
|  |  |  | опанування мови; |  |
|  |  |  | 3 – потрібен спеціальний тренінг з мови; |  |
|  |  |  | 4 – потрібен спеціальний тренінг та |  |
|  |  |  | допомога під час виконання проекту; |  |
|  |  |  | 5 – складно, потрібні лише досвідчені |  |
|  |  |  | люди. |  |
|  |  |  |  |  |

EFactor = 1,5\*4+0,5\*4+1\*3+0,5\*2+1\*1+2\*1-1\*2-1\*3=10;

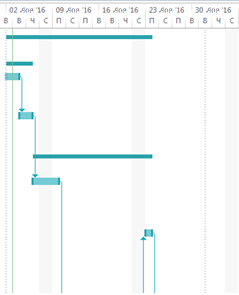
EF = 1,4 + (-0,03 \* EFactor) = 1,4 + (-0,03\*10) = 1,1.

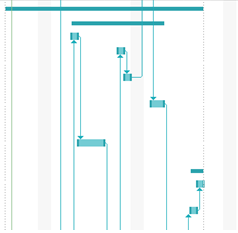
AUCP = UCP \* TCF \* EF = 46\*0,975\*1,1 = 49,335;

Одному UCP відповідає 20 робочих годин.

Отже, загальний час на розробку дорівнює: 20 \* 49,335 = 986,7 (годин).

Діаграма Ганта представлена на рисунку 3.





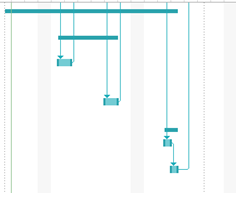


Рисунок 3 – Діаграма Ганта