Міністерство освіти і науки України

Тернопільський національний технічний університет

імені Івана Пулюя

Кафедра комп’ютерних систем та мереж

Звіт

до лабораторної роботи №2

з дисципліни WEB-програмування

на тему:

###### «Проектування архітектури WEB-сайтів»

Виконав:

Студент групи СІ-31

Чабан А.С.

Прийняв:

Яцишин В. В.

Тернопіль 2018

**Тема**: проектування архітектури web-сайтів.

###### Мета: ознайомитись з класифікацією архітектур програмного забезпечення, підібрати найбільш ефективну архітектуру при проектуванні web-сайту, здійснити розбиття на компоненти на найвищому рівні.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Протягом десятиліть розробники програмного забезпечення створювали свої проекти або з нуля, або використовуючи вже накопичений досвід, якщо такий якось вдавалося придбати. В даний час інтенсивно розвивається дисципліна програмної архітектури і проектування. Тепер ми можемо говорити про архітектуру високого рівня і архітектуру низького рівня в термінах, зрозумілих всім професійним розробникам. Точно так, як і ми захоплюємося такими інженерними розробками, як тунель між Великобританією і Францією або проект Міжнародної космічної станції, незабаром ми захоплюватимемося грандіозною програмною архітектурою.

***Архітектура програми***

Якщо порівнювати розробку програм з процесом споруди моста, то аналіз: вимог буде аналогічний вибору місць, де міст починатиметься і закінчується, а також визначенню роду навантажень, які він повинен витримувати. Далі, архітектор моста повинен вирішити, яким буде міст: підвісним, консольним або якогось іншого типу, що задовольняє вимогам. Іншими словами він повинен буде визначити архітектуру моста. Розробники програмною забезпечення стикаються з схожим вибором. У цьому розділі ми розглянемо вибір архітектури для додатку.

Створення архітектури - це проектування на найвищому рівні. Частину процесу проектування що залишилася ми називатимемо детальним проектуванням.

Зрозумілий опис архітектури дуже важливий для всіх застосувань і обов’язково у тому випадку, коли до розробки залучається велика кількість людей. Причиною цього служить необхідність розбиття всього застосування на частини (модулі) з їх подальшою збіркою. Вибір архітектури забезпечує вимагаєму модульність. Розробники, яким доручається створення архітектури (технічні архітектори), зазвичай є найдосвідченішими в команді розробки, всім вимогам необхідно привласнити пріоритети. Приведемо приклад списку основних цілей розробки.

* Розширення.
  + Полегшення додавання нових властивостей.
* Зміни.
  + Полегшення зміни вимог.
* Простота:
  + простота розуміння;
  + простота реалізації.
* Ефективність:
  + досягнення високої швидкості: виконання і (або) компіляції;
  + досягнення малого розміру: об'єктної коди і (або) початкової коди.

Розширення визначає ступінь, в якому архітектура повинна підтримувати додавання нових можливостей в додаток. Найчастіше чим краще архітектура пристосована до розширення, тим більше складну структуру вона має і більше часу потрібно на розробку. Розширюваність зазвичай вимагає введення вищих абстракцій в процес. Наприклад, ми можемо побажати, щоб архітектура нашої відеоігри Зустріч підтримувала не тільки цю гру (нижній рівень універсальності), але і взагалі будь-яку ролеву відеогру. Універсальність дає безліч переваг, але її реалізація вимагає великих витрат часу. Одному з важливих завдань при виборі ступеня універсальності є визначення класу можливих розширень. Ми не можемо проектувати з розрахунку на всі можливі розширення. У зв'язку з цим дуже корисні необов'язкові і бажані вимоги, оскільки вони показують, в яку сторону буде напрямлено розвиток додатку.

Розробка з розрахунком на зміни переслідує інші цілі, хоча має на увазі застосування тих же прийомів проектування, що і для забезпечення розширюваності. В даному випадку ми хочемо спроектувати архітектуру так, щоб вона допускала зміну вимог, наприклад щоб вимогу «гравець повинен постійно мати повний контроль над своїм персонажем» можна було замінити вимогою «гравець час від часу випадковим чином втрачає контроль над своїм персонажем».

Простота є метою проектування за будь-яких обставин. Проста архітектура, яка допускає розширення і зміни, є рідкістю, і її створення вимагає великих зусиль. До інших критеріїв, використовуваних при виборі архітектури, відносяться економія машинного часу і економія пам'яті.

Кількість модулів високого рівня повинна бути невелика. Кількість (норматив), що зазвичай рекомендується, - 7±2, але через специфіку деяких проектів це число може сильно варіюватися. Різниця між дрібномасштабними і великомасштабними проектами вимірюється кількістю рівнів вкладення в модулях. У великомасштабних застосуваннях модулі високого рівня розбиваються на підмодулі, ті у свою чергу розбиваються на під-подмодулі і так далі Норматив 7±2 застосовується для кожної такої декомпозиції.

Як приклад розглянемо декомпозицію гри Зустріч. Один варіант - розбити всі частини гри на чотири модулі.

* Навколишнє середовище, в якому відбувається дія гри (зони, з'єднання і т. д.).
* Механізм управління грою (лічильники, реакції на події і т. д.).
* Учасники гри (персонаж гравця, інші персонажі і т. д.).
* Артефакти, задіяні в грі (мечі, книги, щити і так далі - вони з'являться в майбутніх версіях гри).

Всі ці модулі володіють достатньою зв'язністю. Наприклад, герої гри досить інтенсивно взаємодіють між собою. З іншого боку, зчеплення цих модулів сильніше, ніж нам би хотілося. Наприклад, при зустрічі персонажів задіюються і навколишнє середовище, і механізми управління, і самі персонажі, і артефакти.

Хід роботи

1. Для реалізації проекту обрано архітектуру клієнт-сервер із незалежними компонентами, оскільки вона найкраще підходить до тематики та сфери використання програмного продукту.

2. Декомпозицію системи було здійснено відповідно до трьохрівневої архітектури з такими рівнями:

* Рівень доступу до даних
* Рівень представлення даних.
* Рівень вигляду

Визначившись з темою сайту, можна виділити такі основні пункти навігації користувацької частини сайту (рис. 1).

Система навігації сайту

Новини

Головна сторінка

Слайдер

Головна

Наші проекти

Опис проектів

Послуги

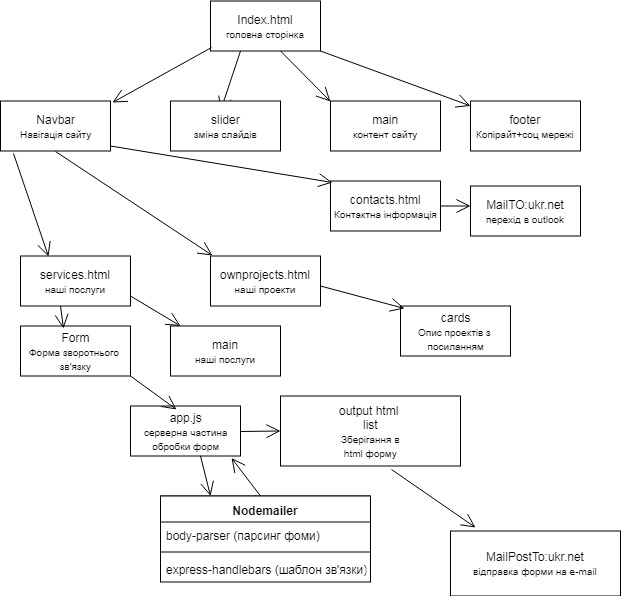
Форма для замовлення п по

Перелік послуг

Контакти

Контактна ііінформація

Рисунок 1. - Система навігації сайтом



Для реалізації наступного завдання розробимо модель сайту

Висновок: Під час виконання даної лабораторної роботи було розглянуто класифікації архітектур програмного забезпечення та підібрано найбільш ефективну архітектуру при проектуванні web-сайту. Також було здійснено розбиття на компоненти на найвищому рівні.