

Національний технічний університет України
“Київський політехнічний інститут ім І. Сікорського”

Звіт

з лабораторної роботи №2
на тему “Архітектура”

Виконав: студент групи
ІС-42 Коростильов Євгеній

Київ 2017 р

1. Поняття про архітектуру

Для початку визначемо що таке архітектура програмного забезпечення. Згідно з Вікіпедією:

Архітектура програмного забезпечення - це структури високого рівня програмної системи, дисципліна створення таких структур та документація цих структур. Ці структури необхідні для роз'яснення щодо програмної системи. Кожна структура містить елементи програмного забезпечення, відносини між ними та властивості як елементів, так і відносин. Архітектура програмної системи є метафорою, аналогічною архітектурі будівлі.

Архітектура програмного забезпечення полягає в тому, щоб зробити фундаментальний структурний вибір, який буде дорого змінювати після того, як буде виконано.

Вибір архітектури програмного забезпечення включає специфічні структурні варіанти від можливостей при розробці програмного забезпечення. Наприклад, системи, що керували ракети-носіями космічного шатуну, мали вимогу бути дуже швидкими та дуже надійними. Тому необхідно вибрати відповідну мову обчислень у реальному часі. Крім того, щоб задовольнити потребу в надійності можна було б зробити вибір, щоб мати кілька надлишкових і самостійно вироблених копій програми, і запустити ці копії на незалежній апаратній в той час як результати перехресної перевірки.

Погоджуючись з цим підемо далі і виберемо архітектуру для проекту. Так як в мене невеликий проект, то не слід використовувати великі масивні архітектури. Тому я вирішив використати мікросервісну архітектуру. Хабр підказує що

Мікросервіси - сучасне уявлення сервіс-орієнтованої архітектури (SOA), що використовується для створення розподілених програмних систем. Як і в SOA, модулі в архітектурі мікросервісов взаємодіють по мережі один з одним для виконання мети. Ще одна схожість в тому, що мікросервіси використовують протокол-незалежну технологію. Дана архітектура є першою реалізацією SOA, що з'явилася після впровадження DevOps, і вона поступово стає стандартом для безперервно розвиваючих систем.

2. Вибір архітектури

Погоджуючись з цим підемо далі і виберемо архітектуру для проекту. Так як в мене невеликий проект, то не слід використовувати великі масивні архітектури. Тому я вирішив використати мікросервісну архітектуру. Хабр підказує що

Мікросервіси - сучасне уявлення сервіс-орієнтованої архітектури (SOA), що використовується для створення розподілених програмних систем. Як і в SOA, модулі в архітектурі мікросервісов взаємодіють по мережі один з одним для виконання мети. Ще одна схожість в тому, що мікросервіси використовують протокол-незалежну технологію. Дана архітектура є першою реалізацією SOA, що з'явилася після впровадження DevOps, і вона поступово стає стандартом для безперервно розвиваючих систем.

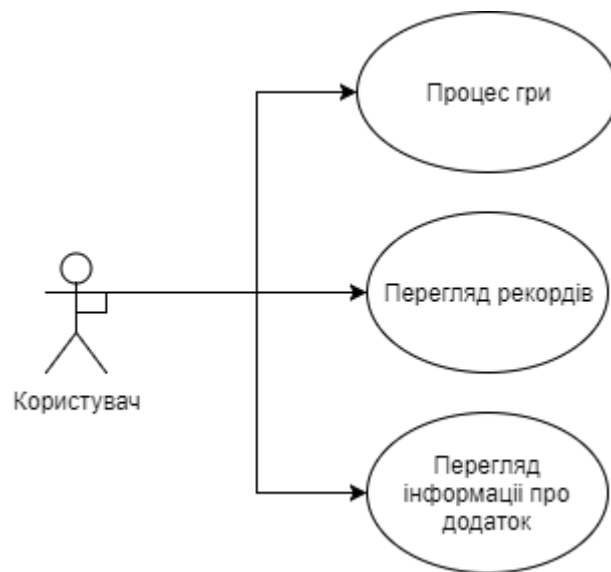
Як ми бачимо така система є дуже зручною для нашого випадку, хоча і не використовуються основні її переваги(розбиття великого на маленьке). Зручніше за все на даному етапі залишити монолітну систему, але для подальшого правильного зростання системи ми вибрали цей тип.

3. Створення мінімального прототипу

Для цього ми зробимо UML-діаграму.

Так як наш додаток є невеликим, то і функцій в ньому у перших версіях буде небагато.

Для кращого читання і розуміння проекту я скоротив функції виведені у діаграму.



UML-діаграма

Як можна бачити з діаграми і з цілей на проект цей малюнок повністю відповідає тому прототипу який в нас був спочатку. Усі функції закладені в систему відображаються повністю.

4. Висновки

На цьому етапі ми дослідили що таке архітектури. Визначили які типи архітектури бувають. Після повного розуміння речей ми визначили яка найбільш підходить для нашого проекту і вибрали її. На основі архітектури ми зробили діаграму в якій всі функції програми відображені повністю.