УДК 004.05

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ КОНВЕРТАЦІЇ ФОРМАТІВ НА БАЗІ TELEGRAM BOT API

В.С. Посвістак, магістрант
Київський національний університет технологій та дизайну
Д.В. Мірошниченко, магістрант
Київський національний університет технологій та дизайну
Т.І. Демківська, кандидат технічних наук, доцент
Київський національний університет технологій та дизайну

Ключові слова: сервер, смартфон, інтернет, мобільний застосунок.

Клієнт-серверна архітектура відіграє важливу роль у розробці ПЗ. На сьогодні вона використовується у більшості програмних продуктів через те, що вирішує проблему побудови комунікації різних модулів кінцевого продукту.

Метою даного дослідження є розробка Telegram Bot та ознайомлення з клієнт-серверною архітектурою, що включає в себе аналіз проблем, які вона вирішує, в першу чергу питання комунікації програмних модулів. Розглянуті можливі альтернативи та приклади використання архітектури в типовому застосунку, що потребує централізоване джерело даних. Такі застосунки широко представлені на сучасному ринку програмного забезпечення. Існує тенденція до зберігання даних в “хмарі” (cloud), тобто не на локальному накопичувачі даних, а на віддаленому. В такому випадку комунікація відбувається через мережу Інтернет. Окрім зберігання даних стали можливі хмарні обчислення, де роль клієнта полягає лише у делегуванні певних операцій на сервер і отримання результатів обчислення. В цих випадках саме за допомогою клієнт-серверної архітектури відбувається побудова кінцевого продукту.

Клієнт-серверна архітектура складається з двох видів програмних модулів – клієнт та сервер. Під терміном клієнт мається на увазі пристрій користувача, тобто персональний комп’ютер, смартфон тощо. Сервер зазвичай знаходиться в іншому місці, зазвичай це більш потужна обчислювальна техніка. Комунікація між клієнтом та сервером відбувається через мережу інтернет.

Класична клієнт-серверна архітектура складається з двох модулів, клієнт та сервер. Інший розповсюджений спосіб проектування системи – трьохрівнева архітектура, що містить в собі:

– клієнт, з яким взаємодіє користувач;

– сервер, що має бізнес-логіку застосунку – всю реалізовану функціональність, якою може користуватись клієнт;

– ресурс-менеджер, що зберігає дані, тобто БД та інтерфейс для комунікації з нею.

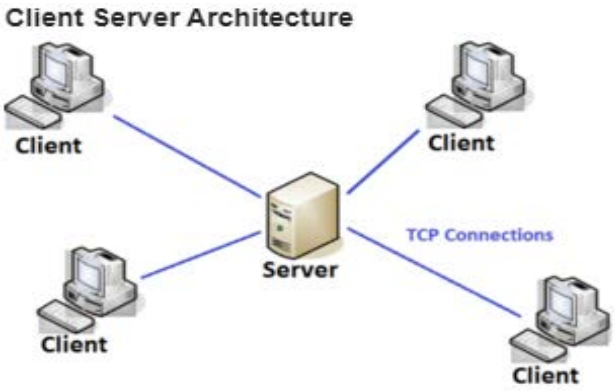

Рисунок 1 – Структурна схема клієнт-серверної архітектури

Програмний модуль-клієнт надає інтерфейс, що дозволяє пристрою робити запити до серверу та показувати результат в будь-якому вигляді. Наприклад, це може бути запит на отримання всіх вхідних повідомлень, та їх відображення у вигляді списку. В даному випадку, сервер буде відповідальний за витягання інформації з бази даних та генерування відповіді для клієнта. Найчастіше клієнт отримує відповідь від сервера у форматі JSON, який є універсальним для об’єктів даних.

Характеристики клієнт-серверної архітектури:

клієнт та сервер потребують різну кількість апаратних та програмних ресурсів, клієнтські та серверні машини можуть належати до різних постачальників;

горизонтальна масштабованість (збільшення клієнтських машин) та вертикальна масштабованість (міграція на більш потужний сервер або на багатосерверне рішення);

один комп’ютер серверного класу може пропонувати одночасно декілька послуг; для кожної послуги потрібен окрема серверна програма в окремому потоці;

Особливістю клієнт-серверною архітектури є направленість запитів під час комунікації. Клієнт призначений для запитів, а сервер для реагування на запити та генерацію відповіді для клієнта. Сервер в цьому випадку є централізованою машиною, від якої залежать усі клієнти. У випадку клієнт-серверної архітектури сервер не має можливості ініціювати запити до клієнта.

В даній роботі проведено дослідження клієнт-серверної архітектури, розглянуто приклади використання та можливі альтернативи для комунікації пристроїв. Проаналізувавши різні випадки, була підтверджена актуальність даної архітектури та виявлені місця де її можна замінити на більш продуктивну. Клієнт-серверна архітектура необхідна в тих ситуаціях, коли необхідне централізоване джерело даних та можна обійтися направленістю комунікації “Запит-відповідь”.

Розроблено TelegramBot— застосунок на мові Kotlin, що дозволяє проводити стандартні маніпуляції з даними: виконуватизапити на отримання необхідних даних, конвертувати файли визначених типів на стороні серверного застосунку.