MANIPAL UNIVERSITY
JAIPUR

A Major Project Report on

SELL MY MOBILES MIOCRSERVICES PROJECT

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELORS OF TECHNOLOGY

In Computer Science and Engineering

2018-2019

By

Dishant Thakur

159102038

Under the guidance of

Dr. Ashish Jain

Department of Computer Science and Engineering

School of Computing and Information Technology

Manipal University Jaipur

Jaipur, Rajasthan

Introduction

Microservices is a form of service-oriented architecture style (one of the most important skills for Java developers) wherein applications are built as a collection of different smaller services rather than one whole app. Instead of a monolithic app, you have several independent applications that can run on their own and may be created using different coding or programming languages. Big and complicated applications can be made up of simpler and independent programs that are executable by themselves. These smaller programs are grouped together to deliver all the functionalities of the big, monolithic app.

While designing systems in microservices architecture, we should be identifying independent components/modules appropriately. These components will be mini applications, which will be developed separately. They will follow their own development and deployment lifecycle.

In this project we are developing one selling mobiles management system. In this system we have various important components like customer management, search management, cart management, order management and review management services.

When we develop this application using microservices architecture we will have independently deployed mini applications for like customer management, search management, cart management, order management and review management .In a general setup we can have scenarios where we need data from various components for a single request. Ideally, we will have an API gateway or front controller which will aggregate data from these components and give it back. We should have inter-component communication. Components can communicate over REST APIs.

Motivation

One of the biggest advantages of the microservices pattern is that it does not require you to rewrite your whole application from the ground up. Instead what you can do is to add new features as microservices, and plug them into your existing application.

Statement of Problem

PROBLEMS OF EXISTING SYSTEM

Monoliths are built as a **single unit**, so they are responsible for every possible functionality: handling HTTP requests, executing domain logic, database operations, communication with the browser/client, handling authentication and so on. Because of this, even the **smallest changes** in the system involves **building and deploying the whole application**. Building and deploying is not the only **problem** - just think about **scaling**. You have to run multiple instances of the monolith, even if you know that the bottleneck lays in one component only.

Methodology

Microservices or microservices architecture, is a software designing technique. It is an architectural style which structures an application as a collection of loosely-coupled services. It has many benefits, like improving modularity and making developers life easy by making developing, testing, and debugging steps easy. It also helps in CI/CD.

One of the biggest advantages of the microservices pattern is that it does not require you to rewrite your whole application from the ground up. Instead what you can do is to add new features as microservices, and plug them into your existing application.

The main steps to be followed to achieve the creation of such a system are:

1. Taking and understanding the requirements of the client/s.
2. Analyzing the design and UI/UX of the potential system.
3. Setting up the working environment for the project.
4. Getting started with coding of the backend/frontend.
5. Unit testing of the backend/frontend.
6. Integrating the whole system together.
7. Integration testing of the whole system.
8. Client review and feedback.
9. Accommodation of any changes whatsoever, and deployment of the system.

System Requirements

Operating system	Windows 8.1 64 Bit, Windows 8 64 Bit, Windows 10 64 Bit, Mac OSX
CPU	Intel Core i5, Intel Core i7 or better
Memory	8 GB RAM

Software Requirements

- Spring Boot
- Post Man
- Swagger
- Jenkins
- Kubernetes
- Eureka

References

- https://dzone.com/articles/basic-things-to-setup-an-microservices-architectur

- websystique.com/spring-boot-tutorial/

- https://www.tutorialspoint.com/spring_boot/index.htm