ABSTRACT

Web application technologies are rapidly evolving, with new features and improvements being added all the time. We'll look at how the popular Spring Boot java-based platform for building online and corporate applications gives service-oriented architecture flexibility (SOA). The level of configuration complexity in any Spring-based application is a hurdle. Spring Boot simplifies the creation and deployment of stand-alone, production-ready Spring apps with minimal Spring configuration. If we use the Spring Model-View-Controller (MVC) framework as an example, we'll need to set up a dispatcher servlet, web jars, a view resolver, and component scan, among other things. To address this, Spring Boot offers a number of Auto Configuration options for configuring the application with any required dependencies. In comparison to CRUD web applications, Spring Boot simplifies dependency management by combining a complete yet flexible framework and supporting libraries into a single dependency that contains all of the Spring-related technologies you'll need for beginning projects.

Keywords: Spring Boot, Backend, Spring Framework, Java, Postman, MVC Architecture.

I. INTRODUCTION

Spring Boot's main or primary feature is autoconfiguration (Simply described, Spring Boot autoconfiguration is a method of automatically configuring a Spring application based on the dependencies found on the class path). By eliminating the need to define some beans that are part of the auto-configuration classes, autoconfiguration can speed up and simplify development. We're focused on several technologies needed to construct an EE (Enterprise Edition) development project, such as Postman for testing written Api's without the requirement for a front-end, MySQL database, Jira for project management utilising Kanban and scrum workflows, and Mockito for writing test cases. Writing many Api's that are commonly used in many EE projects, as well as writing unit tests to test them with Junit5 and Mockito.

Scope and Proposed Model

Mainly Spring is a framework that is built on Java. There are a lot of large apps that are built with the spring framework. Spring technology, which is based on Java, is widely used. These tech-combination projects may be found almost anywhere. All of these issues have been addressed by the Spring framework. This framework develops corporate applications using many innovative approaches such as Aspect-Oriented Programming (AOP), Plain Old Java Object (POJO), and dependency injection (DI), reducing the complexity associated with designing enterprise apps using EJB. Spring Boot aids in the development of stand-alone applications. Because there is less source code with Spring Boot, testing is easier.

II. LITERATURE SURVEY

Spring Framework

Prior to the introduction of Enterprise Java Beans (EJB), Java developers had to create Web apps using JavaBeans. Despite the fact that JavaBeans aided in the building of user interface (UI) components, they were unable to provide services such as transaction management and security, which were required for the development of robust and secure corporate applications. The introduction of EJB was viewed as a solution to this issue. EJB extends Java components like Web and enterprise components and provides services that aid in the building of corporate applications. Developers were looking for a simpler way to construct enterprise applications as a result of this complication. All of these issues have been addressed by the Spring framework.
This framework develops corporate applications using many innovative approaches such as Aspect-Oriented Programming (AOP), Plain Old Java Object (POJO), and dependency injection (DI), reducing the complexity associated with designing enterprise apps using EJB. Spring is a lightweight open-source framework that enables Java EE 7 developers to create easy, dependable, and scalable corporate applications. This framework is primarily concerned with offering multiple methods for managing your business items. When compared to traditional Java frameworks and Application Programming Interfaces (APIs), such as Java database connection (JDBC), Java Server Pages (JSP), and Java Servlet, it makes developing Web applications considerably easier. The Spring framework, which includes Spring AOP, can be thought of as a collection of sub-frameworks, often known as layers. Object-Relational Mapping in Spring (Spring ORM). Spring Web Flow and Spring Web MVC are two of the most popular Spring Web frameworks. When building a Web application, you can use any of these modules separately. To improve the functionality of a Web application, the modules can be joined together.

**Spring Boot**

Spring Boot is constructed on top of the spring and contains all of the spring's characteristics. And is quickly becoming a developer favorite since it provides a quick production-ready environment that allows developers to focus on the logic rather than the configuration and setup. Spring Boot is a microservice-based framework that requires relatively little time to turn into a production-ready application.

**Backend**

The backend (also known as "server-side") is the part of the website that is not visible to the user. It’s in charge of storing and organizing data, as well as ensuring that everything on the client side works properly. The backend and frontend communicate, sending and receiving data that is presented as a web page. Your browser sends a request to the server-side whenever you fill out a contact form, key in a web URL, or make a purchase (any user interaction on the client-side), and the server-side responds with information in the form of frontend code that the browser can read and display.

**Java**

Concurrent execution allows you to run many statements simultaneously rather than sequentially. A programming language that is both class-based and object-oriented. Independent programming language that follows the "Write once, run anywhere" philosophy, which means that compiled code can run on any platform that supports Java. Java is a programming language that may be used to create software for a variety of platforms. It is the most extensively used programming language for Android smartphone apps, independent desktop apps, and networked web applications. Java is a programming language designed exclusively for object-oriented programming. A Java program is converted to bytecode by the Java compiler. The bytecode can be interpreted into machine code by the Java virtual machine, which works on a variety of platforms including Mac, PC, and Unix machines. JIT compilers compile bytecode into native machine code "just in time" for execution, resulting in better JVM performance. It's simple to use, dependable, secure, and platform agnostic.

**IntelliJ**

IntelliJ IDEA is a Java-based Integrated Development Environment (IDE) that aims to boost developer productivity. It takes care of the mundane and repetitive duties for you, such as code completion, static code analysis, and refactoring, allowing you to focus on the more exciting aspects of software development, making it not only productive but also pleasurable.

**Mockito**

Unit testing is one of the methods used by developers to ensure that individual units or components function properly. We need to add the Mockito-core dependency to our build configuration file in order to inject Mockito Mocks into Spring Beans. Mockito is a JAVA-based mocking framework that is used for successful unit testing of JAVA applications. Mockito is a tool for mocking user interfaces so that dummy functionality may be created and utilized in unit testing. This tutorial should teach you how to utilize Mockito's APIs and write unit tests in a straightforward and clear manner. Mockito is a JAVA-based mocking framework that is used for successful unit testing of JAVA applications. Mockito is a tool for mocking user interfaces so that dummy functionality may be created and utilized in unit testing.
Postman

Postman is one of the most widely used program testing tools for API testing. Designers may easily create, test, share, and report APIs with the help of this device. This tutorial will help you understand why Postman is so popular and what makes it unique when compared to other API testing tools. All of the instances in this tutorial have been tested and can be imported into Postman.

III. SYSTEM ARCHITECTURE

The terms REST and API stand for Representational State Exchange and Application Program Interface, respectively. REST is a programming style that defines the set of principles that should be followed while creating web administrations. Calm web administrations are those that follow the REST building pattern. It uses a standard and predetermined set of rules to allow frameworks to access and control online assets. The Hypertext Transfer Protocol (HTTP) is used to communicate in REST-based frameworks (HTTP). A Tranquil framework consists of a client who requests assets and a server who provides them.

Data Flow Diagram

Diagram showing the flow architecture and data flow diagram.
A program is divided into three key cohesive components using the Model-View-Controller (MVC) architecture: Model, View, and Controller. As a result, MVC has been coined. Each design element was built with a specific application enhancement in mind. The commercial rationale and the introduction layer are separated by MVC. MVC was originally intended for graphical user interfaces on desktop computers (GUIs). MVC design is now well acknowledged in web innovation for producing online apps and flexible systems.

IV. SYSTEM DESIGN

Use-case Diagram

Among the UML diagrams that comprise actors and use cases, use case diagrams are the most fundamental. There are three actors in the following use case diagram: the user, the administrator, and the database. The use case diagrams provide a variety of use cases that the actors can accomplish. The registered user can read, modify, and remove their profile. They can also create a profile if one does not exist (CRUD).

Class Diagram
Because they are the only case diagram that can be instantly transformed to object-oriented languages, class diagrams are extensively employed in the design of object-oriented systems. They're used to describe the connections between entities. In the diagram above, we can see a one-to-many relationship between the customer and the bank, as well as a many-to-one relationship between the account and the bank. In the class diagrams, we also include variables and methods that are present in the class.

**Activity Diagram**

Here is a workflow diagram of a user's activity diagram, in which the user enters credentials and the credentials are validated. If the user credentials are present in the database and he or she is not a disabled user, the user is logged into the application and can do various profile functions; if they are not valid, the end point is reached.

**Sequence Diagram**
The sequence diagram displays the user's sequence of actions, account database, and app dashboard based on the actions taken by the user, such as login/signup and performing functionalities in the application.

V. SYSTEM IMPLEMENTATION

![Postman interface to test APIs](image1.png)

**Figure 1:** Postman interface to test APIs

We undertake API integration testing with the help of Postman. API testing is one of the most crucial responsibilities in integrating the front end with the backend. After the APIs has completed the desired functionality, the frontend and backend integration is completed.

![IntelliJ code structure](image2.png)

**Figure 2:** IntelliJ code structure

In our application, here is a sample Java code structure for retrieving user profiles. We are leveraging the Java programming language to create our functionality, as well as various spring boot ideas such as dependency injection and the use of various annotations to achieve our goal.
VI. RESULT AND OUTPUTS

Before hitting API (update API -- PUT)

After hitting API (update API -- PUT)

Before hitting API (delete API -- DELETE)

After hitting API (delete API -- DELETE)
VII. CONCLUSION

The following are the conclusions based on the processes used in this study:

1) When creating a microservice application, the functional requirements should be split down into numerous microservices, which are then integrated to make a coherent application based on defined business processes.

2) As indicated, the Spring Boot application is running on localhost on port 8080, and the constructed REST WS is accessible by the URL mapping "/api/v4". The results show that numerous maven dependencies can be successfully integrated into a Spring Boot application. The developed platform is appropriate for use in a variety of applications that require some artificial inelegance. Because it may be utilized by a variety of clients, realization in the form of a REST WS has several advantages.

3) More functional demands and microservices within a microservice architecture can be added without affecting existing microservices, which is one of the advantages of employing microservice architecture. As a result, the developer can add more functionality on their own, which adds value to the business and saves time for future development.

VIII. REFERENCES


