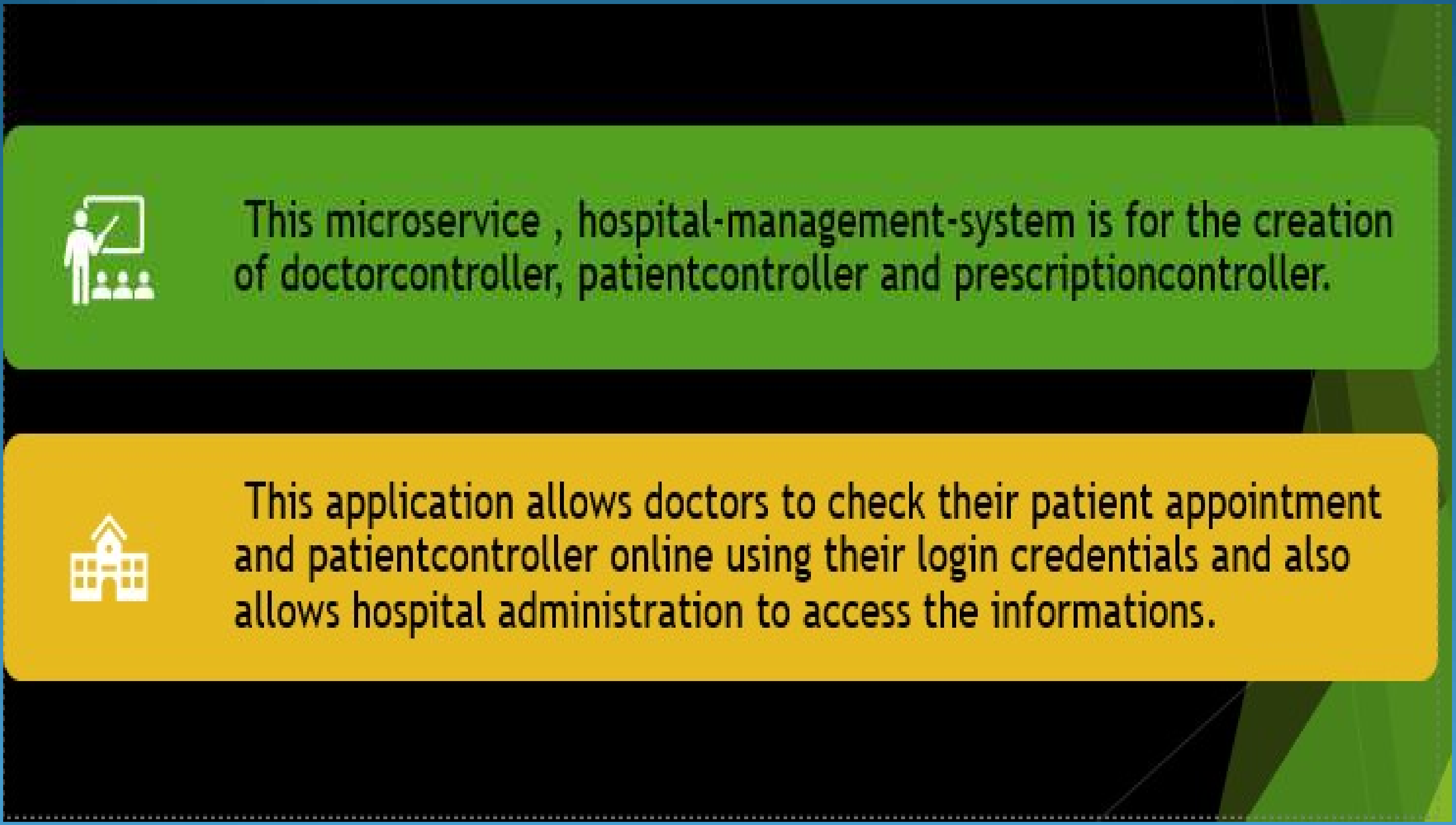
Hospital Management System (HMS)

Submitted By-
Ashish Maurya

Introduction

Hospital Management System is a system enabling hospitals to manage information and data related to all aspects of healthcare – appointment, prescription, patient's informations, and more.

Project Briefing



This microservice , hospital-management-system is for the creation of doctorcontroller, patientcontroller and prescriptioncontroller.

This application allows doctors to check their patient appointment and patientcontroller online using their login credentials and also allows hospital administration to access the informations.

- In this microservice, we are using three controllers and two repository
- In this microservice, first need to long in using login credentials as a patient , doctor or user.
- This microservice can be access data using the patient ,doctor or user only.
- These are the credentials to login. DOCTOR - doctor1 PASSWORD - password
PATIENT - patient1 PASSWORD - password
DOCTOR,PATIENT - user1 PASSWORD – password

- This microservice allows the patient to see prescription and appointment.
- This microservice can manage data by patients, doctors or users credentials only.
- This microservice allows the doctors and user to see and edit prescription and appointment.
- There are three controllers , doctor controller, patient controller and prescription controller .
- There are two repository, Appointmentrepository and prescriptionrepository.

Tools Used

- Apache Maven
- Docker
- Java
- Spring Boot
- MongoDB
- Postman

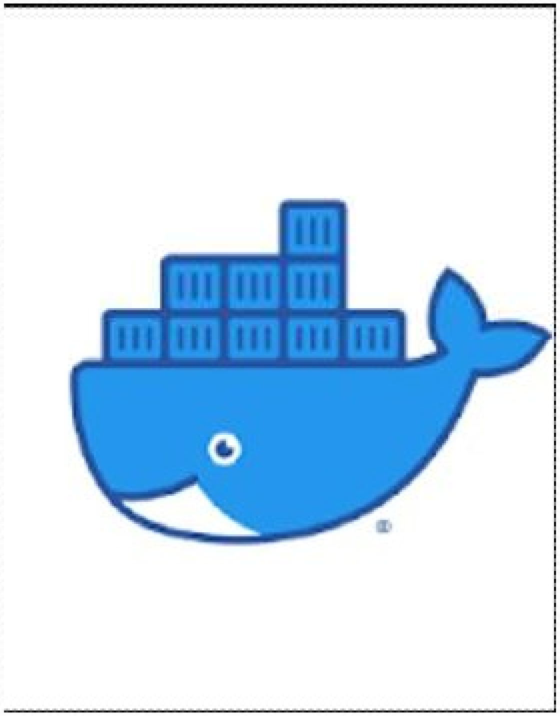
- Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.
- In case of multiple development teams environment, Maven can set-up the way to work as per standards in a very short time. As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups.

Maven provides developers ways to manage the following –
Builds
Documentation
Reporting
Dependencies
SCMs
Releases
Distribution
Mailing list

Java

- Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java.

Docker



- Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

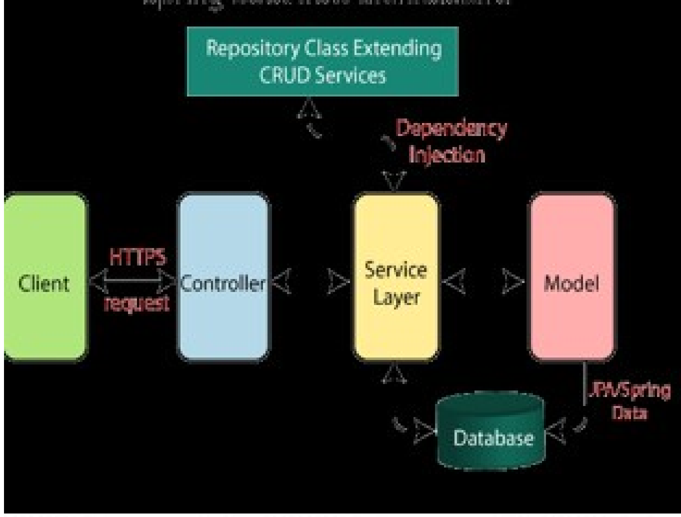
What is Spring Boot?

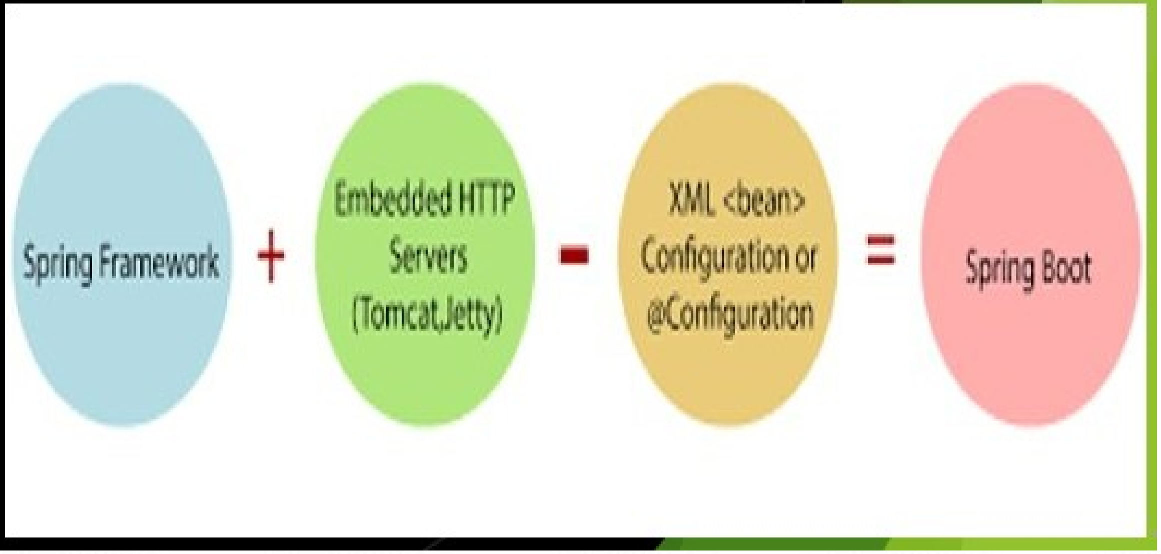
- Spring Boot is an open-source Java based framework used to create a micro service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

Spring Boot Main Components

- Spring Boot Starters- combine a group of common or related dependencies into single dependency
-
- Spring Boot AutoConfigurator- reduce the Spring Configuration
-
- Spring Boot CLI- run and test Spring Boot applications from command prompt
-
- Spring Boot Actuator- provides EndPoints and Metrics

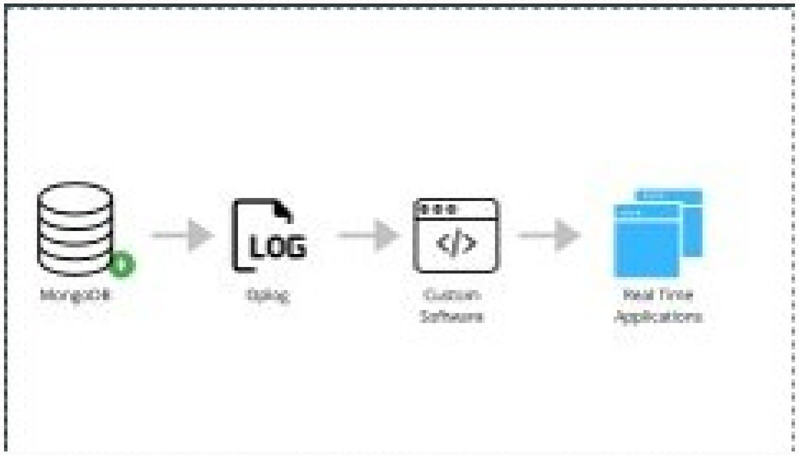
spring®


Spring Boot flow architecture


Spring Framework + Embedded HTTP Servers (Tomcat,Jetty) - XML <bean> Configuration or @Configuration = Spring Boot

MongoDB

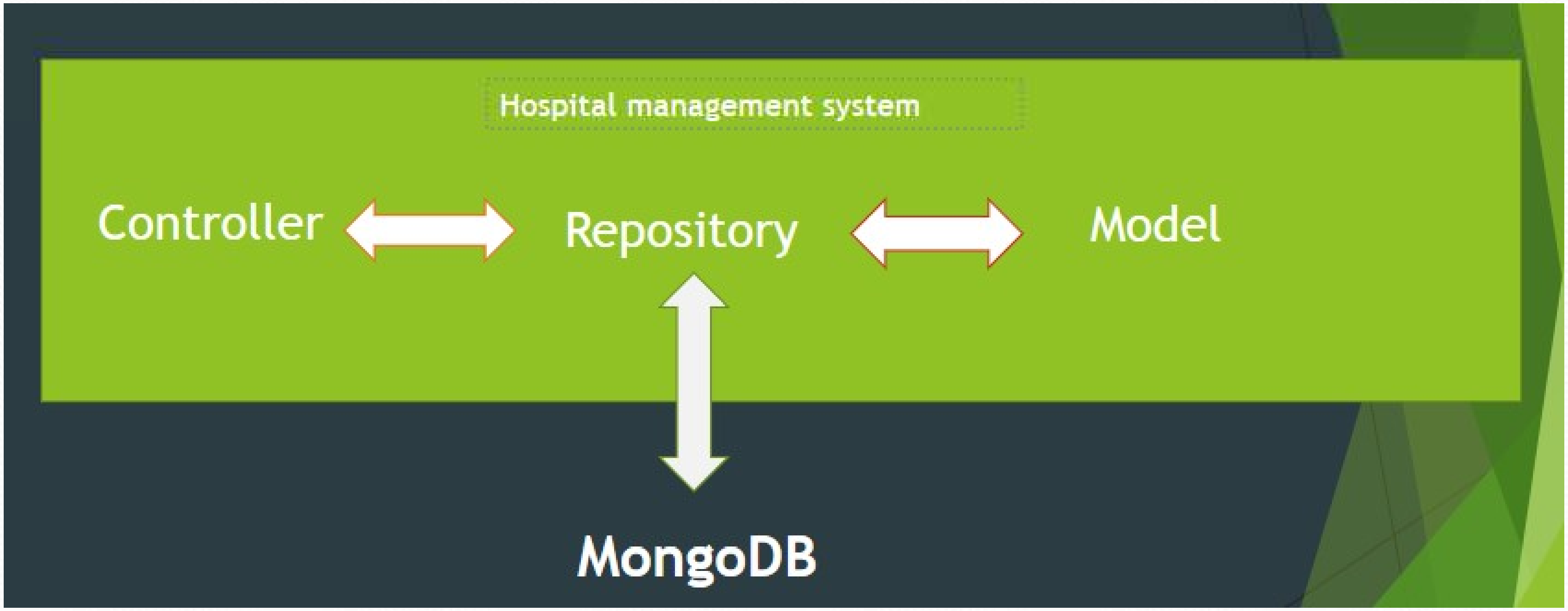
MongoDB is a NoSQL database which stores the data in form of key-value pairs. It is an Open Source, Document Database which provides high performance and scalability along with data modelling and data management of huge sets of data in an enterprise application. MongoDB also provides the feature of Auto-Scaling.



Postman

Postman is an API platform for developers to design, build, test and iterate their APIs

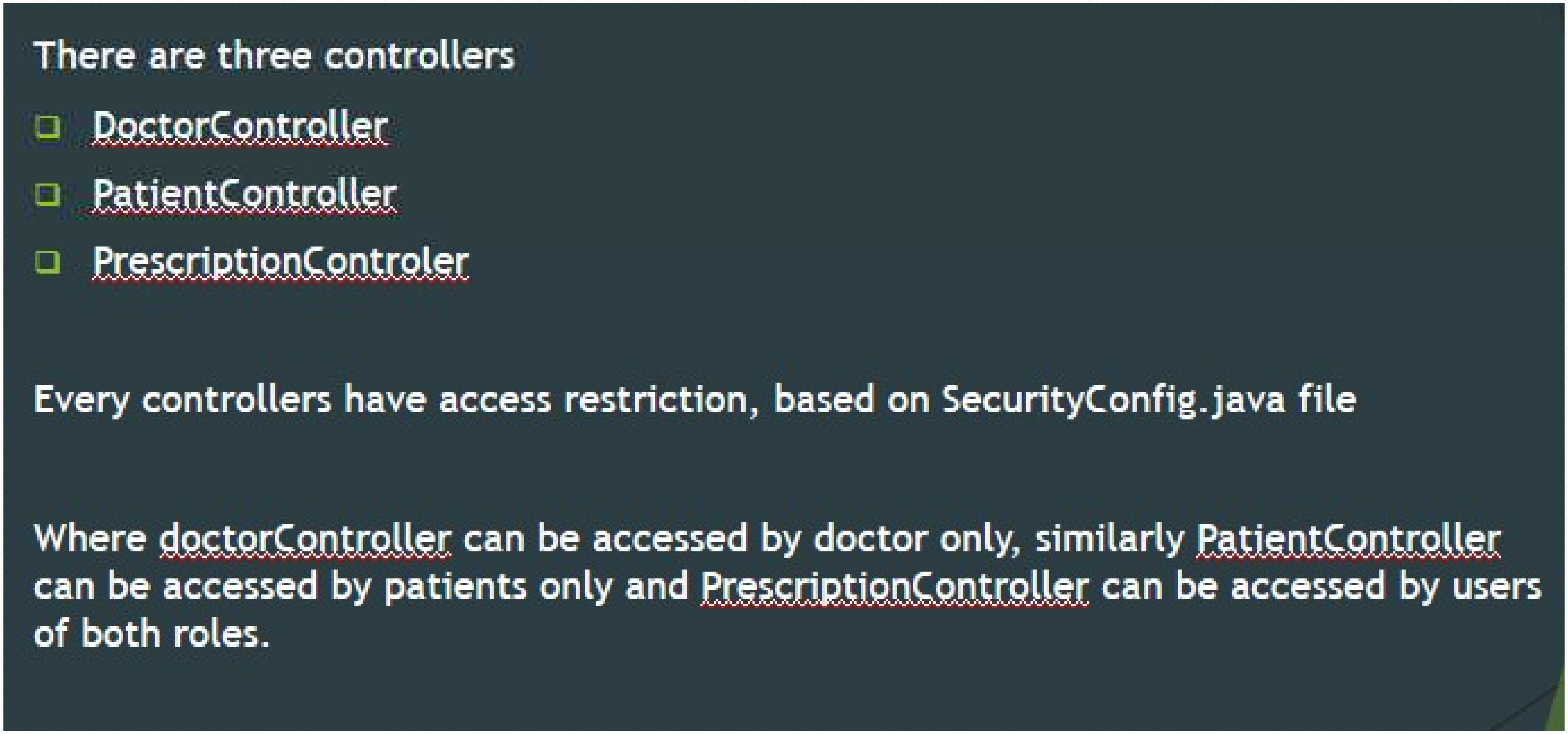
Architecture



Use Cases

- **USE CASE –1** Hostpital may to have record of all the appointment as well as prescription given by doctor alongside patient details.
- **USE CASE –2** For patient to keep record of all his/her appointment as well as prescription.
- **USE CASE –3** For getting patient record so that proper prescription can be provided by the considering all previous prescription

Controllers



There are three controllers

- DoctorController
- PatientController
- PrescriptionControler

Every controllers have access restriction, based on SecurityConfig.java file

Where doctorController can be accessed by doctor only, similarly PatientController can be accessed by patients only and PrescriptionController can be accessed by users of both roles.

- "**/doctorappointment**" - which will GET requests along with request parameters.
- "**/save**" - which will receive POST request along with appointment object in JSON format as Body.
- "**/viewprescription**" - which will receive GET request along with request parameters.
- "**/saveprescription**" - which will receive POST requests along with prescription object in JSON format as body.

Training and code coverage

Test cases were made for each mapping using Mockito and junit with code coverage.

Code coverage:-
Classes – 100%
Methods – 91%
Lines – 88%

Documetation and Dockerization

- Swagger is used for interactive API documentation of the microservice amd its architecture.

- Dockerization is the process of packing ,deploying and running application using Docker containers. Docker is a tool that ships application with all the necessary functionalities as one package.

Thank
You