Report SpringApp Project Group 4

Jacek Antoni Wegrzynowski, Rashaad Wells Iversen, Veronicha C. T. Pettersen

November 18, 2023

Abstract

10-15 lines with the software technology and the highlights from the project that has been undertaken.

1 Introduction

Approximately 1 page on:

- A brief introduction to the prototype implementation and topic of the project. In this project, an IoT-cloud software system has been developed. The system takes feedback from users in the form of yes/no votes on polls. The polls can be voted on either via an web interface or via an IoT-device.
- Discuss (briefly) the technology stack that has been selected, mention related technologies (if relevant), primary arguments for choice of technology stack.

  Here we can maybe add an illustration from the powerpoint
- A brief account of the results that have been obtained in the project.
- A one paragraph overview at the end, explaining how the rest of the report is / has been organised

This rest of this report is organised as follows: Section 2 gives an ....

2 Software Technology Stack

Introduce in (sufficient) depth the key concepts and architecture of the chosen software technologies. As part if this, you may consider using a running example to introduce the technology.

Emphasize the "new" software technologies that was selected by the group and which has not been covered in the course.

This part and other parts of the report probably needs to refer to figures. Figure 1 from [**?**] just illustrates how figure can be included in the report.

2.1 Angular:

The front-end of the FeedApp is developed using Angular. Angular is a widely used framework that is used for building single-page applications (SPAs).

Fundamental Consepts of Angular:

Components and Views : The angular application is built upon components. For every component, there is a HTML template for the content of the web interface and a TypeScript class that controls the logic. The components are what defines the views, which is what is displayed to the user on the webpage.

Dependency Injection (DI) : Angular´s dependency injection (DI) system provides services to components. Services are classes that can contain business logic, data handling and functionalities that can be useful in multiple components. In our FeedApp implementation, we created services dedicated to managing authentication and handling poll data. These services made it possible to streamline data interactions and logic across different components, making it easier to control that the underlying operations and management of the data were handled consistently.

Routing : The Angular Router handles the navigation between different views as users performs different tasks. This is a key element in SPAs since instead of reloading the page

Resourses used for writing this paragraph: https://angular.io/guide/architecture

2.2 Spring Boot

We have chosen Spring Boot as our enterprice software framework when developing our application. Spring Boot is an extension of the Spring Framework that simplifies the development process, making it possible to create a functioning web application fast. To fully understand the benefits of the Spring Boot extension, we are first going to talk a little bit about some of Springs main consepts.

Spring Configurations Explained:

Bean Definitions : In Spring, objects managed by the Spring Inversion of Control (IoC) container are referred to as beans. Configurations involve specifying these beans and managing their lifecycle within the application.

Dependency Injection (DI) : Spring's DI mechanism manages dependencies among application components. This setup is crucial for injecting required services or modules into different parts of the application.

Aspect-Oriented Programming (AOP) : Configurations in Spring also include setting up aspects for handling cross-cutting concerns like logging or transaction management.

Data Source and Transaction Management : For applications interacting with databases, configurations encompass setting up database connections and managing transactions effectively.

Deploying a Spring Application:

Packaging : The application is compiled and packaged, typically into a JAR or WAR file, ready for deployment.

Running on a Server : The packaged application is deployed on a web server. Spring Boot, with its embedded server capability, simplifies this by allowing the application to run independently without needing a separate server setup.

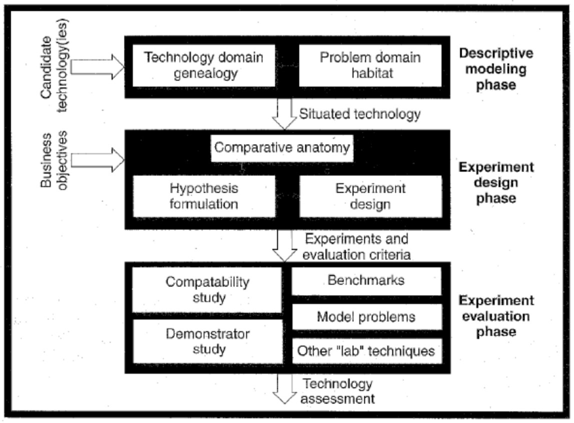

Figure 1: Software technology evaluation framework.

Spring Boot's Role in FeedApp:

Auto-Configuration : Spring Boot automatically configures the application based on the included libraries, reducing the need for extensive manual configuration.

Simplified Deployment : The embedded server feature of Spring Boot allows our application to be deployed as a standalone unit, enhancing ease of deployment and portability.

In the implementation of our FeedApp prototype, the use of SpringBoots autoconfiguration and deployment functionalities has meade it possible for us to spend more time on the applications buissness logic.

2.3 JSON Web Tokens

2.4 H2

2.5 Hibernate

2.6 Java Persistence API

2.7 Mosquitto MQTT

3 Demonstrator Prototype

About 4 pages on:

1. An architectural overview of the application that has been implemented
2. High-level design, domain model, ... (App assignment A)
3. May involve selected models from Chaps. 5 of the IoT and cloud books

The example below shows how you may include code. There are similar styles for many other langages - in case you do not use Java in your project. You can wrap the listing into a figure in case you need to refer to it. How to create a figure was shown in Section 2.

Config	Property	States	Edges	Peak	E-Time	C-Time	T-Time
22-2	A	7,944	22,419	6.6 %	7 ms	42.9%	485.7%
22-2	A	7,944	22,419	6.6 %	7 ms	42.9%	471.4%
30-2	B	14,672	41,611	4.9 %	14 ms	42.9%	464.3%
30-2	C	14,672	41,611	4.9 %	15 ms	40.0%	420.0%
10-3	D	24,052	98,671	19.8 %	35 ms	31.4%	285.7%
10-3	E	24,052	98,671	19.8 %	35 ms	34.3%	308.6%

Table 1: Selected experimental results on the communication protocol example.

```java
public class BoksVolum {

   public static void main(String[] args) {

      int b, h, d;
      String btext, htext, dtext;

      [ ... ]

      int volum = b * h * d;

      String respons =
           "Volum [" + htext + "," + btext + "," + dtext + "] = " + volum;

   }
}
```

4 Prototype Implementation

This section should provide details of how the prototype has been implemented which may involve presentation of suitable code snippets.

5 Test-bed Environment and Experiments

About 2 pages that:

Explains how the prototype has been tested the test-bed environment.

Explains what experiments have been done and the results.

For some reports you may have to include a table with experimental results are other kinds of tables that for instance compares technologies. Table 1 gives an example of how to create a table.

6 Conclusions

Concludes on the project, including the technology, its maturity, learning curve, and quality of the documentation.

The references used throughput the report should constitute a well chosen set of references, suitable for someone interesting in learning about the technology.