**Факултет техничких наука у Чачку**

**Универзитета у Крагујевцу**

**Никола Радовановић**

**Spring Boot framework и његова архитектура на примеру чет апликације**

**дипломски рад**

**Чачак, октобар 2022.**

**Факултет техничких наука у Чачку**

**Универзитета у Крагујевцу**



**Spring Boot framework и његова архитектура на примеру чет апликације**

**Дипломски рад**

**Врста студија: Основне академске студије**

**Назив студијског програма: Информационе технологије**

**Предмет: Напредно објектно-оријентисано програмирање**

**Студент: Руководилац рада:**

**Никола Радовановић 54-2017 Др Олга Ристић, ванр. проф.**

**Чачак, октобар 2022.**

**Резиме**

Програмирање је делатност која постоји преко седамдесет година и захтева константо напредовање технологија које се користе. Иако технологије константно напредују, основе програмирања су махом остале исте откако се појавио први објектно оријентисани језик.

Java програмски језик је своју популарност стекао захваљујући својој портабилности, односно не завишношћу од система који је покреће. Због тога се разноврсност примене Jave креће од једноставних десктоп апликација, преко мобилних апликација до сложених веб сервера.

Због популарности Jave и зато што постоје програмска решења за проблеме са којима се програмери често сусрећу, појављује се Spring фрејморк. Spring je фрејмворк за јава програмски језик који помаже да се програмери фокусирају на нова решења тако што обезбеђује комплетну инфраструктуру апликације. Другим речима програмери се могу фокусирати на проблемске захтеве саме апликације уместо да се прво баве израдом инфраструктуре.

Инфрастуктура коју Spring обезбеђује укључује и конекцију путем интернета са другим апликацијама кроз веб сокет. Као и протокол за слање порука

Кључне речи:

* Java
* Spring
* Spring Boot
* Фрејмворк
* Веб сокет

**Summary**

Keywords:

* Java
* Spring
* Spring Boot
* Framework
* Web socket
* Application

# Садржај

[1. Увод 5](#_Toc114775636)

[2. Spring Framework 6](#_Toc114775637)

[2.1 Креирање Spring boot пројекта 6](#_Toc114775638)

[3. План развоја апликације 7](#_Toc114775639)

[3.1 Коришћени алати и програмска подршка 7](#_Toc114775640)

[3.1.1 Spring Boot 7](#_Toc114775641)

[3.1.2 Maven 7](#_Toc114775642)

[3.1.3 Websocket 7](#_Toc114775643)

[3.1.4 STOMP 7](#_Toc114775644)

[3.1.5 InteliJ 7](#_Toc114775645)

[3.1.6 JavaScript 7](#_Toc114775646)

[3.1.7 JPA 7](#_Toc114775647)

[3.2 Коришћене зависности 7](#_Toc114775648)

[3.3 Глобални опис 7](#_Toc114775649)

[3.3.1 Перспектива система 7](#_Toc114775650)

[3.3.2 Карактеристике система 7](#_Toc114775651)

[3.4 Дефинисање захтева апликације 7](#_Toc114775652)

[3.4.1 Функционални захтеви 7](#_Toc114775653)

[3.4.2 Нефункционални захтеви 9](#_Toc114775654)

[3.4.3 Процесни захтеви 10](#_Toc114775655)

[4. Архитектура апликације 11](#_Toc114775656)

[4.1 Spring Boot архитектура 11](#_Toc114775657)

[4.2 Архитектура веб сокета 11](#_Toc114775658)

[4.3 STOMP архитектура 11](#_Toc114775659)

[4.4 База података 11](#_Toc114775660)

[4.5 Генерална архитектура 11](#_Toc114775661)

[5. Изглед апликације 12](#_Toc114775662)

[6. Закључак 13](#_Toc114775663)

[7. Закључак 14](#_Toc114775664)

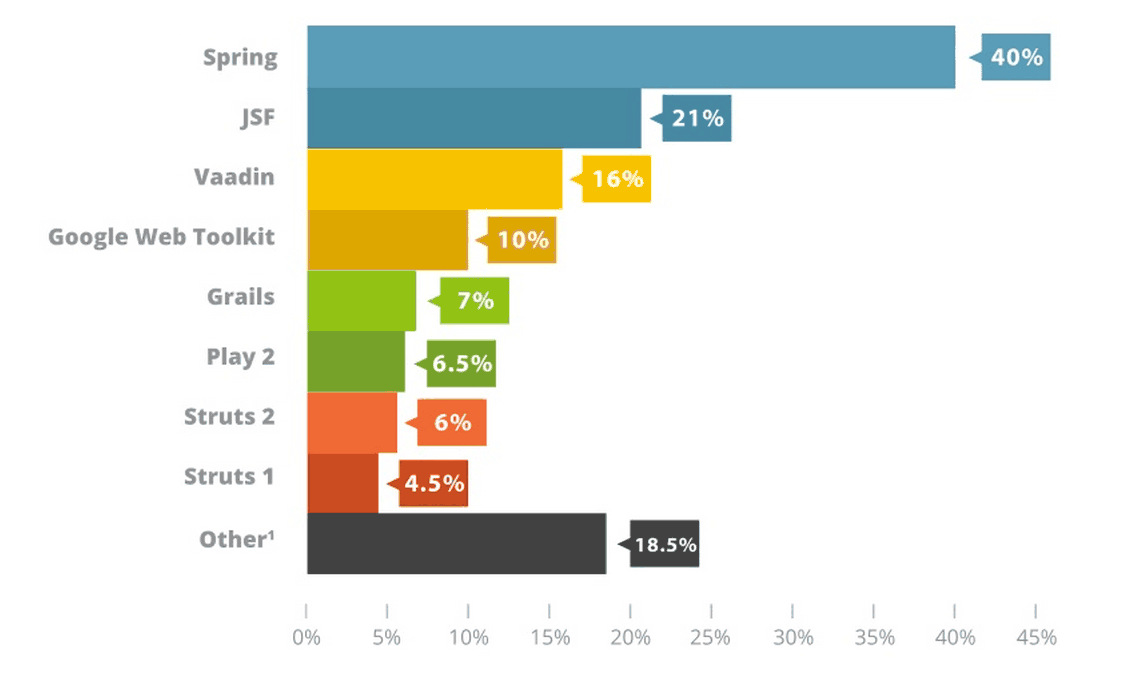
[8. Списак слика 15](#_Toc114775665)

# Увод

За развијање комплексних софтвера потребно је увести доста компоненти у саму структуру апликације, као што су на пример: База података, модул који се повезује са осталим апликацијама путем интернета, контролери... Како би се смањило време израде оваквих апликација појавила се потреба за софтвером који ће олакшати повезивање свих делова апликације. Готово за сваки програмски језик постоји фрејмворк. У листи испод приказани су неки од најпопуларнијих фрејмворка као и њихови програмски језици.

1. ASP .NET CORE (C#)
2. Bootstrap (HTML, CSS)
3. Rails (Ruby)
4. Laravel (PHP)
5. Angular (JavaScript)
6. Spring (Java)

За потребе овог рада креирана је апликација у Јава програмском језику уз помоћ Spring фрејмворка. Spring није једини фрејмворк за Јаву, али је убедљиво најпопуларнији због своје флексибилности, скалабилности и лакоће коришћења. На слици испод приказанa је популарност јава фрејмворка према броју корисника у процентима (<https://www.jrebel.com/blog/java-web-framework-usage-stats>).



Као што се може приметити, спринг је тренутно најпопуларнији јава фрејмворк. Због својих предности предвиђа се још већи раст његове популарности. Неке од ових предности су:

* Смањује време израде софтвера, а повећава ефикасност.
* Лако повезивање са базама података
* Интеграција са Spring екосистемом што омогућава коришћење Spring Data, Security, ORM и JDBC
* Уштеда времена при писању кода

Циљ дипломског рада је да прикаже архитектуре спринг боот фрејмворка, веб сокета, стомп протокола и генералну архитектуру апликације, на примеру апликације за групно дописивање. Апликација је поприлично једноставне функционалности, али се у позадини дешава много више ствари.

# Spring Framework

Спринг се први пут помиње у књизи Рода Џонсона ,,*Expert One-on-One J2EE Design and* 2002. године. Род је развио Спринг како би олакшао развој серверских јава апликација. Спринг је пуштен у рад 2003. године под лиценцом Apache 2.0.

Спринг боот настаје десетак година касније тако што је Мајк Јангстром на форуму предложио унапређење спринг-а кроз обједињење микросервиса. Глсвни покретач овог пројекта је Фил Веб из Спринг развојног тима, који прву верзију спринг боот-а објављује 2014. године

## Разлика између спринг и спринг боот фрејмворка

Спринг је најпопуларнији фрејмворк за развој апликација у Јави. Међутим, током последњих неколико година, Спринг је постао све сложенији због додатних функционалности. Започињање новог пројекта обично захтева дуготрајан процес и одузима много времена.

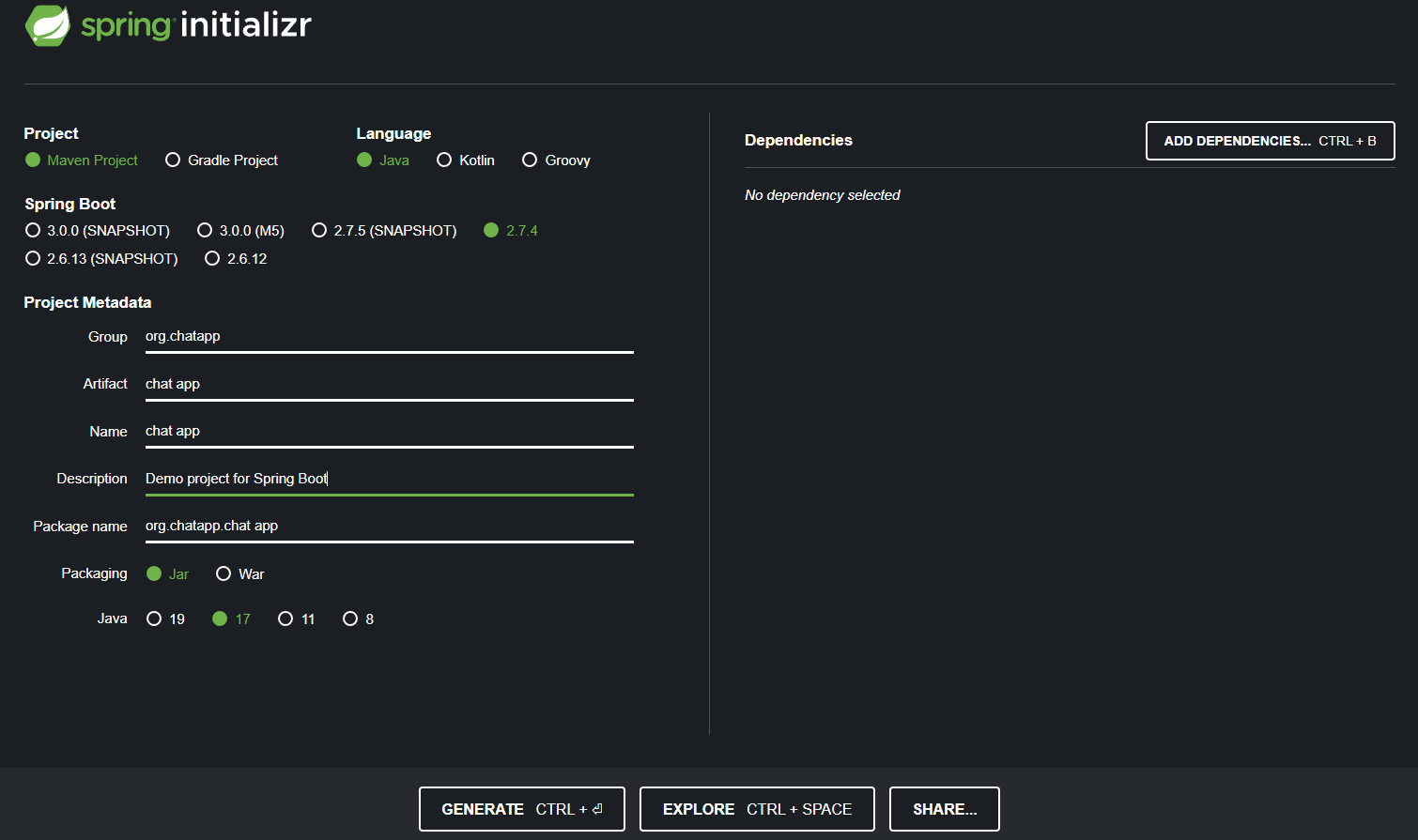
Да би се избегло почињање од нуле и да би се уштедело време, уведен је Спринг Боот. Спринг Боот користи Спринг као свој темељни слој и омогућава програмерима да направе самосталну апликацију без конфигурације или са минималном конфигурацијом. Посебно је усмерен ка изградњи веб апликација и микросервиса.

Спринг боот омогућава лакшу конфигурацију и мање посла око инфраструкруре пројекта. Свакако се базина на основном спринг-у и не може функционисати без њега. За ову апликцаију коришћен је спринг боот верзије 2.7.4.

## Креирање спринг боот пројекта

То ће бити Спринг Боот Мавен пројекат најлакше се генерише уз помоћ Spring Initializr (<https://start.spring.io/>). Ово је најбржи начин за креирање Спринг апликације, а ИнтеллиЈ ИДЕА обезбеђује додатак у оквиру самог софтвера за спринг инитиализр.

У суштини Spring Initializr је веб апликација која аутоматски генерише Spring Boot структуру пројекта за нас. Не генерише код, али се доста смањује посао око конфигурације. На слици 2 приказан је изглед ове веб апликације.



У Project делу дефинише се тип пројекта, затим је потребно одабрати програмски језик. Можемо да бирамо између Jave, Kotlin-a и Groovy. Верзију спринг боот фрејмворка селектујемо испод истоимене секције. Снапсхоот верзије су тестне верзије и нису увек стабилне. Последња стабилна верзија је 2.7.4. Потребно је такође попунити податке о апликацији и одабрати врсту пакета и верзију Јаве.

У депендециес делу се селектују зависности, односно које сервисе желимо додати у апликацију. Најчешће коришћене зависности су: Spring Web, Spring Security, Spring JPA…

Када се заврши са основним конфигурисањем приојекта, све што је преостало јесте да се преузме генерисани пројекат. Пројекат који се преузима је аутоматски повезан са свим селектованим сервисима, на пример са базом података, конфигурсан је да користи интернет ако је додата зависност за веб сокет итд.

# План развоја апликације

## Коришћени алати и програмска подршка

### Spring Boot

### Maven

Мавен је алатка за прављење и управљање пројектима која се обично користи у фрејмворцима изграђеним у Јави. Развила га је Апацхе Софтваре Фоундатион.

Пре него што је Мавен ушао у пројекте, све зависности, које нису ништа друго него ЈАР датотеке, морале су се ручно додавати у фрејмворк. Такође, требали смо се побринути за надоградњу софтвера у нашем пројекту.

Мавеном управља датотека пројектног модела (POM). У овој датотеци се додају све зависности и конфигуришу плагинови који се додају, као и основне конфигурације пројекта попут имена, верзије јаве и пакета.

### Websocket

### STOMP

### InteliJ

### JavaScript

### JPA

## 3.2Коришћене зависности

## 3.3Глобални опис

### Перспектива система

.

### Карактеристике система

## Дефинисање захтева апликације

### Функционални захтеви

*Табела. Функционални захтеви за корисника*

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број | Захтев | Опис | Приоритет |
| 1. | Логовање | Систем треба да обезбеди приступ аутентификованом кориснику | 1 |
| 2. | Додавање ставки на рачун | Корисник може да дода жељене ставке на рачун | 1 |
| 3. | Наплаћивање | Корисник може да изврши наплату | 1 |
| 4. | Генерисање фактуре | Корисник моће да генерише фактуру за извршену продају | 1 |
| 5. | Додавање купаца | Додавање купаца у базу података | 1 |
| 6. | Измена података о купцу | Корисник може да мења податке о купцу | 1 |
| 7. | Преглед и претрага купаца | Корисник може да прегледа и претражује купце који се налазе у бази података | 1 |
| 8. | Преглед и претрага производа | Корисник може да прегледа и претражује производе | 1 |
| 9. | Сторнирање | Корисник може да изврши сторнирање рачуна | 1 |
| 10. | Преглед рачуна | Корисник може да прегледа ставке на рачунима | 2 |
| 11. | Преглед новости | Корисник може да прегледа новости | 2 |

У табели 2. приказани су функционални захтеви за администратора.

*Табела 2. Функционални захтеви за администратора*

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број | Захтев | Опис | Приоритет |
| 1. | Логовање | Систем треба да обезбеди приступ администратору | 1 |
| 2. | Уређивање производа | Администратор може да прегледа, додаје, мења и брише производе | 1 |
| 3. | Уређивање купаца | Администратор може да прегледа, додаје, мења и брише купце | 1 |
| 4. | Уређивање корисника | Администратор може да прегледа, додаје, мења и брише кориснике | 1 |
| 5. | Преглед рачуна | Администратор има увид у рачуне | 1 |
| 6. | Издавање и сторнирање рачуна | Администратор може да издаје и сторнира рачуне | 1 |
| 7. | Додавање обавештења | Администратор може да додаје обавештења која ће бити приказана на почетној страни | 2 |

### Нефункционални захтеви

У табели 3. приказани су нефункционални захтеви.

*Табела 3. Нефункционални захтеви система*

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број | Захтев | Опис | Приоритет |
| 1. | Перформансе | Систем треба да одговара на корисникове захтеве у што краћем року | 1 |
| 2. | Безбедност | Систем треба да забрани приступ неаутентификованим корисницима | 1 |
| 3. | Ограничење приступа | Корисници не смеју да имају приступ појединим деловима система | 1 |
| 4. | Поузданост | Систем треба да буде доступан 24/7 | 1 |
| 5. | Ажурност | Уколико се на систему открију грешке потребно их је уклонити у што краћем року | 1 |
| 6. | Интуитивност | Навигација кроз систем треба да буде што једноставнија | 1 |
| 7. | Windows подршка | Систем ће подржавати Windows оперативне системе | 1 |
| 8. | Скалабилност | Систем ће подржавати приступ више корисника истовремено | 1 |
| 9. | Дизајн | Систем ће имати савремени дизајн | 2 |
| 10. | Подршка за различите прегледаче | Систему ће моћи да се приступи са различитх веб прегледача | 2 |
| 11. | Техничка подршка корисницима | Корисници ће на располагању имати техничку подршку на даљину | 3 |

### Процесни захтеви

У табели 4. Приказани су процесни захтеви.

*Табела 4. Процесни захтеви система*

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број | Захтев | Опис | Приоритет |
| 1. | Програмски језик | Апликација ће бити рађена у Java програмском језику | 1 |
| 2. | Фрејмворк | Апликација ће користити Spring Boot фрејмворк | 1 |
| 3. | База података | Апликација ће користити ЈПА базу података, и у њој ће бити чувани сви подаци | 1 |
| 4. | Генерисање фактура | За генерисање фактура користиће се Jasper Reports алат | 1 |
| 5. | Фронтенд | Фронтенд део система биће кодиран у HTML-у уз подршку Bootstrap-a и JavaScript-a | 1 |
| 6. | Развојно окружење | За развој апликације користиће се InteliJ развојно окружење | 2 |

# Архитектура апликације

## Spring Boot архитектура

## Архитектура веб сокета

## STOMP архитектура

## База података

## Генерална архитектура

# Изглед апликације

# Закључак

# Закључак

1. Spring Boot Reference Documentantion, преузето са: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>, приступано јула 2022.
2. Greg L. Turnquist: Learning Spring Boot 2.0 Second Edition: Simplify the development of lightning fast applications based on microservices and reactive programming, Novembar 2017, 370p.
3. Dinesh Rajput: Mastering Spring Boot 2.0: Build modern, cloud-native, and distributed systems using Spring Boot, Maj 2018, 380p.
4. Joshua Block: Effective Java: Second Edition, 2004, 262p.
5. Learn Spring Boot, преузето са: <https://www.baeldung.com/spring-boot>, приступано јула 202.
6. Spring Boot – Quick Guide, преузето са: <https://www.tutorialspoint.com/spring_boot/spring_boot_quick_guide.htm>, приступано јула 2022.
7. JasperReports Documentation, преузето са: <https://community.jaspersoft.com/documentation?version=61916>, приступано јула 2022.
8. Top 10 Java Frameworks to Use in 2022, преузето са: <https://hackr.io/blog/java-frameworks>, приступано јула 2022.
9. Securing a Web Application, преузето са: <https://spring.io/guides/gs/securing-web/>, приступано јула 2022.
10. Learn Java, преузето са: <https://dev.java/learn/>, приступано јула 2022.

# Списак слика

[*Слика 1. Spring Initializr* 8](#_Toc111817976)

[*Слика 2. Изглед пројекта након импорта* 8](#_Toc111817977)

[*Слика 3. Изглед pom фајла* 10](#_Toc111817978)

[*Слика 4. Изглед Eclipse радног окружења* 12](#_Toc111817979)

[*Слика 5. Spring Boot архитектура* 18](#_Toc111817980)

[*Слика 6. Spring Boot архитектура радног тока* 20](#_Toc111817981)

[*Слика 7. Контролер класа* 21](#_Toc111817982)

[*Слика 8. Entity класа* 22](#_Toc111817983)

[*Слика 9. Пример репозитуријума* 23](#_Toc111817984)

[*Слика 10. Пример сервис класе* 23](#_Toc111817985)

[*Слика 11. Изглед архитектуре унутар пројекта* 24](#_Toc111817986)

[*Слика 12. Структура базе података* 26](#_Toc111817987)

[*Слика 13. Страница са формом за логовање* 27](#_Toc111817988)

[*Слика 14. Изглед почетне стране* 28](#_Toc111817989)

[*Слика 15. Страница за продају* 29](#_Toc111817990)

[*Слика 16. Страница за продају - део за наплату* 29](#_Toc111817991)

[*Слика 17. Изглд генерисане фактуре* 30](#_Toc111817992)

[*Слика 18. Страница за преглед рачуна* 31](#_Toc111817993)

[*Слика 19. Преглед ставки са рачуна* 31](#_Toc111817994)

[*Слика 20. Страница „Artikli“* 32](#_Toc111817995)

[*Слика 21. Страница за измену артикла* 32](#_Toc111817996)

[*Слика 22. Страница „Kupci“* 33](#_Toc111817997)

[*Слика 23. Грешка при отварању „Admin“ странице* 33](#_Toc111817998)

[*Слика 24. „Admin“ страница* 34](#_Toc111817999)