

Informacijski sustav izračuna proizvodnih cijena

Cecelja, David

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:332858>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-23**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)





MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

David Cecelja, 0066096953

**INFORMACIJSKI SUSTAV IZRAČUNA
PROIZVODNIH CIJENA**

Završni rad

Čakovec, prosinac 2023.



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

David Cecelja, 0066096953

**INFORMACIJSKI SUSTAV IZRAČUNA
PROIZVODNIH CIJENA**

COST ESTIMATE INFORMATION SYSTEM

Završni rad

Mentor:

dr. sc. Josip Nađ, pred.

Čakovec, prosinac 2023.

MEDIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 20. veljače 2023.

država: **Republika Hrvatska**
Predmet: **Programsko inženjerstvo i informacijski sustavi**

ZAVRŠNI ZADATAK br. 2022-RAČ-I-91

Pristupnik: **David Cecelja (0066096953)**
Studij: **Izvanredni preddiplomski stručni studij Računarstvo**
Smjer: **Programsko inženjerstvo**

Zadatak: **Informacijski sustav izračuna proizvodnih cijena**

Opis zadatka:

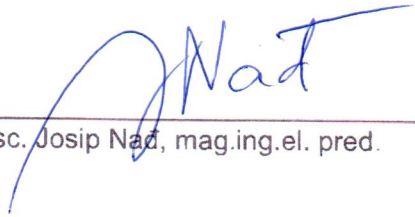
Cilj Završnog rada je opisati principe rada informacijskih sustava za izračun proizvodne cijene poluproizvoda i gotovih proizvoda. Kao osnovicu za prikaz poslovanja treba koristiti tzv. osnovni proizvodni model koji bi predstavljao proizvodno poduzeće srednje veličine.

Na osnovi prikazane ideje, potrebno je definirati osnovne zahtjeve za informacijskim sustavom koji bi podržao traženi način rada, te izraditi mali informacijski sustav s mogućnostima unosa osnovnih količinskih i vrijednosnih podataka koji bi bili osnova za izračun cijena. Informacijski sustav kao završni korak treba izračunati proizvodne cijene poluproizvoda i gotovih proizvoda na osnovu izabranih parametara.

Rok za predaju rada: 20. rujna 2023.

Mentor:

Predsjednik povjerenstva za
završni ispit:



dr. sc. Josip Nađ, mag.ing.el. pred.

ZAHVALA

Želim se zahvaliti svom mentoru na neizmjerljivoj podršci, stručnom vođenju i dragocjenim savjetima koje mi je pružao tijekom izrade ovog završnog rada. Također, želim izraziti veliku zahvalnost svim profesorima na Međimurskom veleučilištu u Čakovcu. Njihova stručnost i predanost omogućili su mi stjecanje neprocjenjivog znanja potrebnog za ostvarenje svih obrazovnih ciljeva.

SAŽETAK

Cilj završnog rada je razviti i dokumentirati informacijski sustav za izračun proizvodnih cijena, s posebnim naglaskom na njegovu primjenu u industrijskoj proizvodnji. Ovaj sustav ima ključnu ulogu u automatizaciji procesa izračuna troškova proizvodnje i formiranju cijena proizvoda, što je od suštinskog značaja za uspješno poslovanje u industrijskom sektoru. Informacijski sustav obuhvaća sve relevantne troškove koji se pojavljuju tijekom proizvodnog procesa, uključujući troškove materijala, rada, energije, opreme i drugih resursa. On je fleksibilan i može se prilagoditi različitim vrstama proizvoda i načinima proizvodnje kako bi zadovoljio specifične potrebe korisnika. U ovom specifičnom slučaju sustav će pratiti troškove proizvodnje preko troškova materijala i strojeva koji se koriste u proizvodnji

Ključni aspekt ovog sustava je kvalitetno definiranje matičnih podataka. To uključuje detaljne informacije o resursima, troškovima materijala i radne snage te druge relevantne informacije koje su ključne za precizno izračunavanje troškova proizvodnje. Nakon što se svi troškovi unesu u sustav, on provodi izračune kako bi se dobili ukupni proizvodni troškovi, koji se potom koriste za formiranje cijena proizvoda.

Ova informacija o proizvodnim cijenama ima široku primjenu u poslovanju. Pomaže u donošenju ključnih odluka o proizvodnji, postavljanju konkurentne cijene proizvoda, planiranju marketinških strategija i drugim aspektima poslovanja. Kroz ovaj sustav, tvrtkama se omogućava učinkovito upravljanje troškovima proizvodnje i sposobnost da postave konkurentne cijene za svoje proizvode na tržištu.

Završni rad detaljno istražuje sve aspekte razvoja i implementacije informacijskog sustava za kalkulaciju proizvodnih cijena, uključujući arhitekturu, dizajn korisničkog sučelja, bazu podataka, testiranje te odabir programskih alata i tehnologija.

Izgradnja informacijskog sustava obuhvaća sučelje za prijavu, registraciju i autorizaciju korisnika s korištenjem Java Spring razvojnog okruženja. Korišteni su programski alati poput Eclipse za pisanje koda i MySQL Workbench za upravljanje bazom podataka. Dizajn sučelja koristi Thymeleaf i Bootstrap za dinamički izgled.

Baza podataka odražava matične podatke iz Excel tablica, koristeći Hibernate za objektno-relacijsko mapiranje. Aplikacija se sastoji od različitih slojeva, uključujući model za klase povezane s bazom, kontrolere za obradu zahtjeva, repozitorije za upravljanje bazom podataka, rukovođenje greškama, konfiguraciju za sigurnost, statičke HTML predloške i korištenje Apache Maven za upravljanje projektom. Za izvođenje aplikacije koristi se Apache Tomcat kao poslužitelj za Java web aplikacije.

Završni rad pruža uvid u proces stvaranja informacijskih sustava za industrijsku proizvodnju, kojima se doprinosi efikasnosti i konkurentске prednosti tvrtki u tom sektoru.

Ključne riječi: *Informacijski sustav, matični podaci, proizvodni troškovi, cijena proizvoda*

ABSTRACT

The goal of the final thesis is to develop and document an information system for calculating production prices, with special emphasis on its application in industrial production. This system plays a key role in automating the process of calculating production costs and forming product prices, which is essential for successful business in the industrial sector. The information system includes all relevant costs that appear during the production process, including the costs of materials, labor, energy, equipment and other resources. It is flexible and can be adapted to different types of products and production methods to meet the specific needs of users. In this specific case, the system will track production costs through the costs of materials and machines used in production

The key aspect of this system is the quality definition of master data. This includes detailed information on resources, material and labor costs, and other relevant information that is critical to accurately calculating production costs. Once all costs are entered into the system, it performs calculations to obtain total production costs, which are then used to form product prices.

This information on production prices has a wide application in business. It helps in making key decisions about production, competitive pricing of products, planning marketing strategies and other aspects of business. Through this system, companies are enabled to effectively manage production costs and the ability to set competitive prices for their products on the market.

The final paper examines in detail all aspects of the development and implementation of an information system for the calculation of production prices, including architecture, user interface design, database, testing and selection of software tools and technologies.

The construction of the information system includes an interface for logging in, registering and authorizing users using the Java Spring development environment. Software tools such as Eclipse for writing code and MySQL Workbench for database management were used. The interface design uses Thymeleaf and Bootstrap for a dynamic look.

The database reflects master data from Excel tables, using Hibernate for object-relational mapping. The application consists of various layers, including a model for database-related classes, request processing controllers, database management repositories, error handling,

security configuration, static HTML templates, and using Apache Maven to manage the project. To run the application, Apache Tomcat is used as a server for Java web applications.

The final paper provides insight into the process of creating information systems for industrial production, which contribute to the efficiency and competitive advantage of companies in that sector.

Keywords: *Information system, master data, production costs, product price*

SADRŽAJ

1. UVOD	10
2. PROIZVODNI TROŠKOVI	11
2.1. Procjena troškova	11
2.2. Matični podaci	11
2.3. Izvještaji	13
2.4. Tipovi aktivnosti	13
3. INFORMACIJSKI SUSTAV	14
3.1. Prikupljanje zahtjeva	14
4. KOMPONENTE INFORMACIJSKOG SUSTAVA	16
4.1. Hardver	16
4.2. Softver	16
4.3. Podaci	16
4.4. Ljudi	16
5. STVARANJE VRIJEDNOSTI	17
5.1. Agilnost	17
5.2. Potreba za Informacijskim Sustavom	18
5.3. Benefiti informacijskog sustava	18
6. ARHITEKTURA INFORMACIJSKOG SUSTAVA	20
6.1. Sigurnost informacijskog sustava	20
6.2. Dizajn informacijskog sustava	20
6.3. Arhitektura	21
6.4. Korisničko sučelje	22
6.5. Baza podataka	22
6.6. Testiranje	23

7. IZGRADNJA INFORMACIJSKOG SUSTAVA	24
7.1. Programski alati.....	24
7.1.1. <i>Eclipse</i>	24
7.1.1. <i>MySQL Workbench</i>	25
7.2. Dizajn sučelja	25
7.2.1. <i>Bootstrap</i>	25
7.3. Autorizacija	26
7.4. Dizajn baze podataka.....	26
7.4.1. <i>Tablice</i>	27
7.4.1. <i>Hibernate</i>	28
8. DIJELOVI APLIKACIJE	30
8.1. Model	31
8.2. Service.....	31
8.3. Kontroler	32
8.4. Repozitorij.....	33
8.5. Rukovođenje greškama i iznimkama	34
8.6. Konfiguracija.....	34
8.7. Statički dio aplikacije	35
8.8. Apache Maven.....	36
8.9. Apache Tomcat	36
9. PRIKAZ SUČELJA I FUNKCIONALNOSTI	37
10. ZAKLJUČAK	44
IZJAVA O AUTORSTVU.....	45
LITERATURA.....	46
POPIS ILUSTRACIJA.....	47
PRILOZI.....	48

1. UVOD

Završni rad se bavi informacijskim sustavom za kalkulacije proizvodnih cijena. Kalkulacija je računski postupak izračunavanja cijena te se kod proizvodnih poduzeća koristi za utvrđivanje troškova proizvodnje samog proizvoda. Dakle, kalkulacije se koriste kako bi se troškovi s glavnih mjesta troškova rasporedili na konačne nositelje odnosno na proizvode. Informacijski sustav za izračun proizvodnih cijena je automatizirani proces izračuna troškova proizvodnje i formiranja cijene samog proizvoda. Ovaj sustav u sebe uključuje sve troškove koji se javljaju u procesu proizvodnje, kao što su troškovi materijala, rada, energije, opreme, alata te se mora prilagoditi specifičnim potrebama korisnika. Nakon što se svi troškovi unesu u takav automatizirani sustav, on izračunava proizvodne troškove i koristi ih za formiranje cijene pojedinog proizvoda. Ove informacije su od presudne važnosti za donošenje odluka o proizvodnji, cijeni, marketingu i ostalim poslovnim aspektima.

Informacijski sustav u proizvodnji sadrži izvještaje o trenutnim troškovima radnih naloga. Izvještaji o trenutnim troškovima sadrže podatke na osnovu proizvodnih knjiženja vrijednosti proizvedenih količina, vrijednosti potrošenih količina materijala i proizvodnih odstupanja. U svrhu gradnje informacijskog sustava potrebno je prikupiti podatke od klijenta i tu se koristi inženjerstvo zahtjeva. Prvo se mora proučiti organizacijska struktura poduzeća. Zatim se definiraju matični podaci koje će novi informacijski sustav koristiti. Nakon toga se definiraju strukturni parametri za izračun troškova. Bitno je definirati poslovne procese kao što su planiranje proizvodnih troškova i praćenje proizvodnih troškova da bi se ti poslovni procesi mogli realizirati. Rad će se baviti izračunom cijene koštanja te praćenjem proizvodnih troškova.

Na kraju prednosti informacijskog sustava za izračun cijena su efikasno upravljanje troškovima proizvodnje, brže i preciznije kalkulacije, analiza i praćenje trendova u troškovima proizvodnje i podrška upravi firme u donošenju poslovnih odluka.

2. PROIZVODNI TROŠKOVI

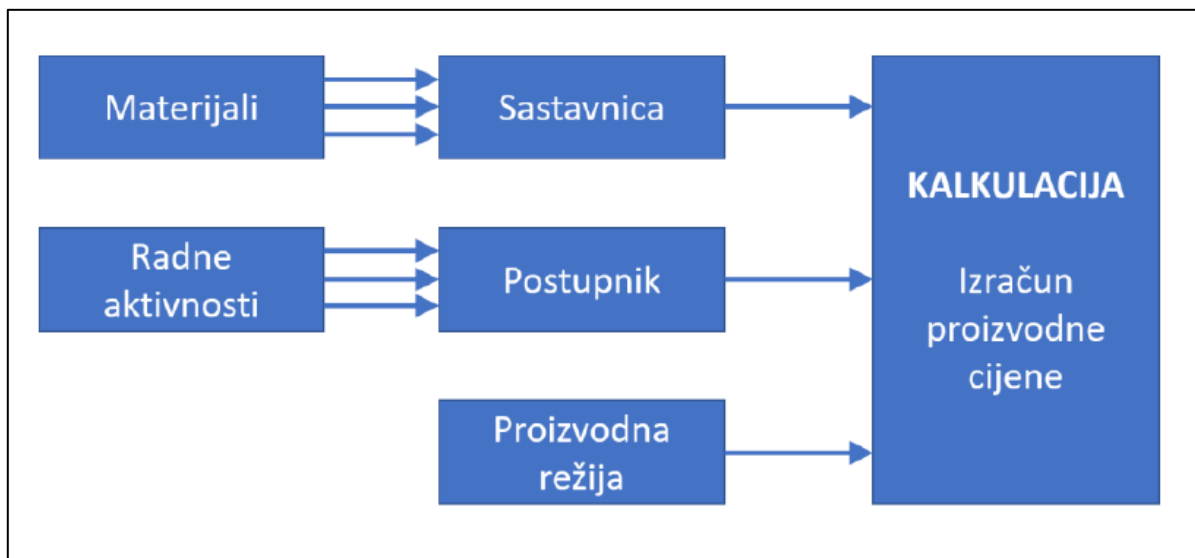
Jedna od najvažnijih komponenti u poslovanju tvrtke je sustav koji omogućuje prikupljanje i analiziranje svih proizvodnih troškova (npr. trošak utrošenih materijala, plaće radnika, amortizacijski troškovi, režijski troškovi). Proizvodni matični podaci koji su neophodni za rad informacijskog sustava izračuna proizvodnih cijena su: parametri u matičnom slogu materijala, sastavnice, mjesta rada i planovi operacija.

2.1. Procjena troškova

Procjena troškova proizvodnje se radi prije početka proizvodnje i kao rezultat se dobiva cijena izrade proizvoda i poluproizvoda. Do cijene se dolazi na osnovu strukture proizvoda, cijena materijala, cijena aktivnosti i ključa općih troškova, a analiza troškova se radi pomoću strukture troškovnih komponenti.

2.2. Matični podaci

Proizvodna cijena se računa na osnovu matičnih podataka. Matični podaci su podijeljeni na sastavnice i postupnik. Sastavnice su popisi materijala koji su potrebni za proizvodnju, a postupnik je popis radnih aktivnosti potrebnih za proizvodnju određenog proizvoda.

Slika 1. Princip kreiranja proizvodne cijene

Izvor: Nađ, J. (2020): Programsko inženjerstvo i informacijski sustavi, Međimursko veleučilište u Čakovcu, Čakovec, str. 29.

Prilikom izračuna se za svaki materijal iz sastavnice i za svaku radnu aktivnost iz postupnika uzima jedinična cijena. Potom se iz sastavnice uzima podatak o potrebnim količinama pojedinog materijala i iz postupnika se uzimaju podaci o količini radnih aktivnosti proizvodnje.

Ovdje se dobiva kalkulacija kojom se bavi završni rad tj. koju informacijski sustav izračunava. Nadalje tu se još dodaje planski trošak proizvodne režije u odnosu na planski trošak materijala i dobivamo rezultat izračuna jedinične proizvodne cijene tj. dobiva se kalkulacija cijene. Stvarni proizvodni troškovi se izračunavaju na osnovi provedenih knjiženja utroška materijala korištenih u proizvodnji i utroška vremena proizvodnje. Troškovi radnih naloga se vode po principu:

- Planskih troškova - izračunavaju se prilikom kreiranja radnog naloga
- Stvarnih troškova - predstavljaju rezultat knjiženja materijala i tipova aktivnosti
- Ciljanih troškova – izračunavaju se prilikom zatvaranja perioda i predstavljaju normativne troškove za stvarno proizvedene količine.

2.3. Izvještaji

Informacijski sustav proizvodnje sadrži dvije vrste izvještaja:

- izvještaji o trenutnim troškovima radnih naloga
- *drill-down*¹ izvještaji temeljeni na posebnom procesu prikupljanja podataka

Izvještaji o trenutnim troškovima sadrže podatke o vrijednosti proizvedenih količina, vrijednosti potrošenih količina materijala i podatke o proizvodnim odstupanjima. Drill-down izvještaji se zasnivaju na hijerarhijskom principu. Podaci koji se mogu analizirati na taj način su planski, stvarni i ciljni troškovi i količine, te sve definirane kategorije proizvodnih odstupanja.

2.4. Tipovi aktivnosti

Za izvođenje interne raspodjele troška potrebno je definirati i tipove aktivnosti kod kojih se aktivnosti mjesta troška koriste kao faktori praćenja troškova. Interna raspodjela troškova po tipovima aktivnosti znači da se određeni troškovi mogu pratiti prema količinama aktivnosti, koje se mogu izmjeriti, zapisati i alocirati u neku bazu podataka.

Za praćenje proizvodnih troškova koriste se radni nalozi. Osnovna ideja se pri tome zasniva na razdvajanju proizvodnih i općih troškova. Troškovi rada ljudi i strojeva dolaze na nalog pomoću potvrđivanja radnog naloga, kada se definiranjem korištenog resursa radi direktna alokacija tipova aktivnosti s proizvodnog mjesta troška. Izlaz radnog naloga je nekakav materijal (proizvod, poluproizvod ili među proizvod). Svi ostali troškovi se svrstavaju u opće (režijske) troškove i kao takvi se na nalog prebacuju na kraju mjeseca.

¹ Proces transformacije podataka u relacijskom modelu podataka bilo po tabličnom ključu ili pivotu.

3. INFORMACIJSKI SUSTAV

Informacijski sustav treba osigurati relevantne informacije korisnicima radi donošenja odluka. Informacijski sustav za izračun proizvodnih cijena koristi različite algoritme i metode za automatsko izračunavanje troškova proizvodnje i formiranje cijene proizvoda. Koristeći informacije o matičnim podacima, njihovu cijenu, količinu, sustav automatski izračunava ukupne proizvodne troškove, a zatim koristi određene formule i modele za formiranje cijene proizvoda.

Personalizacija sustava za specifične potrebe korisnika: Informacijski sustav za izračun proizvodnih cijena može se personalizirati za specifične potrebe korisnika. Na primjer, korisnik može definirati svoje vlastite formule za izračun cijene proizvoda, dodati svoje vlastite troškove ili izmijeniti postojeće modele. Time se osigurava da sustav odgovara specifičnim potrebama svakog korisnika.

3.1. Prikupljanje zahtjeva

Da bi informacijski sustav bio personaliziran prvo se pristupa inženjerstvu zahtjeva. Prilikom izračuna kalkulacije u završnom radu nužno je kvalitetno postaviti zahtjeve za matične podatke.

Slika 2. Princip prikupljanja zahtjeva

Izvor: Nađ, J. (2020): Programsko inženjerstvo i informacijski sustavi, Međimursko veleučilište u Čakovcu, Čakovec, str. 36.

Ovdje će se razgraničiti koje tipove materijala će se koristiti, koji su parametri za materijale, koji su parametri bitni za sastavnice, koji su parametri bitni za postupnike i koji su statusi bitni za pojedine vrste matičnih podataka.

4. KOMPONENTE INFORMACIJSKOG SUSTAVA

Informacijski sustav se sastoji od pet ključnih komponenata: hardvera, softvera, podataka, ljudi i procesa. Prve tri komponente spadaju u tehničku stranu, dok se posljednje dvije komponente odnose na ljudske aspekte.

4.1. Hardver

Hardver, ili računalni hardver, predstavlja fizičke komponente računala koja se koriste u kompaniji. Softver, s druge strane, predstavlja nematerijalni dio računala. Ovo uključuje programe i podatke koji se nalaze na računalima, uključujući operativni sustav.

4.2. Softver

Softver je skup instrukcija koje upravljaju radom hardvera. Razvoj softvera uključuje stvaranje niza uputa koje oblikuju ponašanje računalnog sustava.

4.3. Podaci

Treći dio informacijskog sustava su podaci. Podatke možemo smatrati sirovim informacijama, poput činjenica koje opisuju svojstva objekata i njihove odnose. Podaci postaju informacija kada steknu značaj i postanu relevantni za primatelja, pridonoseći razumijevanju ili znanju primatelja. Organizacije prikupljaju različite vrste podataka i analiziraju ih kako bi donijele informirane odluke, koje se potom mogu procijeniti prema njihovoj učinkovitosti.

4.4. Ljudi

Ljudski faktor igra najvažniju ulogu u informacijskim sustavima. Ovaj aspekt uključuje različite uloge, od radnika na prvoj liniji službe za pomoć do sistemskih analitičara, programera i glavnih informatičkih direktora. Ljudi su odgovorni za korištenje, održavanje i razvoj informacijskih sustava te igraju vitalnu ulogu u njihovom uspjehu.

5. STVARANJE VRIJEDNOSTI

Iskorištavanje tehnologije kako bi se poboljšali poslovni rezultati i doprinijelo organizaciji osnova je stvaranja vrijednosti pomoću informacijskog sustava. Informacijskim sustavom mogu se pojednostaviti poslovi, smanjiti troškovi, ubrzati proces donošenja odluka i učiniti ga jednostavnijim, što zajedno doprinosi stvaranju vrijednosti u kompaniji. Osnovni način stvaranja vrijednosti je pojednostavniti operacije u kompaniji. Baš informacijski sustav služi tome da se ponavljajući zadaci automatiziraju, da se greške uzrokovane ljudskim faktorom svedu na minimum i na kraju da se poveća brzina operacija a samim time i učinkovitost. Na taj način firma smanjuje svoje troškove, produktivnost se povećava kao i kvaliteta usluga.

Još jedan od načina stvaranja vrijednosti pomoću informacijskog sustava je unaprjeđenje donošenja odluka. Sada je organizacijama omogućen pristup stvarnim podacima i analizama u stvarnom vremenu, što olakšava donošenje odluka menadžmentu. Pristup preciznim i ažuriranim informacijama pomaže organizacijama u donošenju odluka koje poboljšavaju njihove poslovne rezultate i potiču rast.

5.1. Agilnost

Korisničko iskustvo je također unaprijeđeno. Razumijevanjem preferencija i ponašanja korisnika omogućuje se da organizacija prilagodi svoje proizvode i usluge prema potrebama svojih kupaca. Još neke od koristi informacijskog sustava su stvaranje vrijednosti preko inovacija prilikom eksperimentiranja, dijeljenja znanja i povećanje agilnosti kompanije. Korištenjem tehnologije, organizacije mogu razvijati nove proizvode, ulaziti na nova tržišta i stvarati nove poslovne modele. Agilnost se odnosi na sposobnost organizacije da brzo reagira na promjenjive tržišne uvjete i poslovne zahtjeve. Ovdje je opet informacijski sustav od ključne važnosti jer svojom brzinom i omogućavanjem brzog donošenja odluka organizacije postaju fleksibilne i prilagodljive te na taj način otpornije na izazove tržišta.

5.2. Potreba za Informacijskim Sustavom

Danas je potreba za informacijskim sustavom u kompaniji neupitna. On djeluje kao centralno spremište podataka i informacija iz različitih izvora i odjela unutar kompanije što olakšava upravljanje podacima i smanjuje rizik od gubitka podataka. Sigurnost podataka je bitna jer čuva integritet kompanije. Pristup podacima je strogo kontroliran i zbog toga je sustav prijave podešen da samo ovlašteni korisnici mogu pristupiti određenim podacima i funkcionalnostima. Time se sprječava neovlašteni pristup i štiti sigurnost podataka. Brz pristup podacima i mogućnost njihove analize u stvarnom vremenu omogućava organizaciji brzo donošenje odluka i održavanju prednosti pred konkurencijom. Informacijski sustav je potreban jer svojom skalabilnošću omogućava rast kompanije. Kompanije rastu s vremenom i to rezultira rastom baze podataka. Dobro osmišljen informacijski sustav može se povećavati ili smanjivati prema potrebama organizacije kako bi se prilagodio rastućem broju korisnika i količine podataka ili njihovom smanjenju prilikom restrukturiranja.

5.3. Benefiti informacijskog sustava

Informacijski sustav kalkulacije proizvodnih cijena može donijeti niz benefita za krajnje korisnike, bilo da se radi o malim poduzećima ili velikim korporacijama. Mogući benefiti su:

1. Precizna procjena troškova: Sustav omogućuje detaljno praćenje svih komponenti proizvodnih troškova, uključujući sirovine, radnu snagu, energiju i opremu. To pomaže korisnicima da precizno procijene cijene proizvoda ili usluga.
2. Optimizacija proizvodnih procesa: Informacijski sustav može analizirati podatke o proizvodnji kako bi identificirao učinkovite ili neefikasne procese. To korisnicima omogućuje optimizaciju proizvodnih operacija i smanjenje nepotrebnih troškova.
3. Bolja upravljačka odlučivanja: Na temelju preciznih podataka, korisnici mogu donositi informirane odluke o cijenama proizvoda, količinama proizvodnje i marketingu. To dovodi do boljih strategija za rast i profitabilnost.
4. Smanjenje rizika: Informacijski sustav omogućuje praćenje promjena u cijenama sirovina, poreznim zakonima i drugim faktorima koji utječu na cijene proizvodnje. Kao rezultat toga, korisnici mogu bolje upravljati rizicima i prilagoditi svoje poslovanje promjenama na tržištu.

5. Povećanje konkurentske prednosti: Korištenje naprednih informacijskih sustava za kalkulaciju proizvodnih cijena može pomoći poduzećima da se izdvoje na tržištu nudeći konkurentne cijene i visoku kvalitetu proizvoda.

6. Bolja kontrola nad zalihama: Sustav može pratiti razine zaliha sirovina i gotovih proizvoda te automatski upozoriti korisnike o potrebama za dopunjavanjem zaliha ili smanjenju prekomjernih zaliha.

7. Poboljšana transparentnost: Korisnici mogu bolje razumjeti strukturu svojih troškova i kako se oni odražavaju na cijene proizvoda. Ovo može poboljšati internu transparentnost i komunikaciju unutar organizacije.

8. Ušteda vremena i resursa: Automatizacija procesa kalkulacije proizvodnih cijena smanjuje potrebu za ručnim radom i minimizira pogreške u proračunima, što štedi vrijeme i resurse.

9. Povećana kvaliteta proizvoda: Bolja kontrola nad proizvodnim procesima omogućuje korisnicima da osiguraju višu kvalitetu svojih proizvoda, što može povećati zadovoljstvo kupaca i lojalnost.

U konačnici, Informacijski sustav kalkulacije proizvodnih cijena može značajno doprinijeti boljem upravljanju poslovanjem, povećanju konkurentske prednosti i povećanju profitabilnosti za krajnje korisnike.

6. ARHITEKTURA INFORMACIJSKOG SUSTAVA

Cilj programskog inženjerstva je stvoriti produktivan i kvalitetan informacijski sustav koji ispunjava sve zahtjeve klijenta. Tijekom razvoja se mogu pojaviti određeni izazovi koje onda rješava tim programskih inženjera i developera u suradnji sa menadžerima kompanije za koju je predviđen informacijski sustav. Svaki informacijski sustav ima svoje nedostatke i ograničenja, a ponekad su ti nedostaci izraženiji ovisno o specifičnostima implementacije. Jedan od najčešćih nedostataka je visok trošak implementacije. Razvoj i implementacija informacijskog sustava mogu biti financijski izazovni, uključujući troškove softvera, hardvera, korisničke edukacije i kasnijeg održavanja sustava.

Sustavi, osobito oni koji podržavaju složene poslovne procese, mogu postati izuzetno složeni, što dodatno otežava njihovo održavanje i nadogradnju i povećava troškove. Jedan od izazova je ovisnost o specifičnim tehnologijama i platformama na kojima je informacijski sustav sazidan. Promjene u tim tehnologijama zahtijevaju dodatne prilagodbe sustava, što može biti skupo i vremenski zahtjevno.

6.1. Sigurnost informacijskog sustava

Sigurnost je još jedan izazov jer informacijski sustavi su često meta sigurnosnih prijetnji poput krađe podataka ili virusa u svrhu industrijske špijunaže. Na kraju je izazov edukacije korisnika za korištenje novog sustava i tu ljudski faktor igra veliku ulogu naročito je tu bitna uloga Menadžera za promjene kojemu je zadatak pospješiti prijelaz sa staroga na novi sustav. Uvođenje novih sustava može izazvati otpor kod osoblja. Ako sustav nije fleksibilan i prilagodljiv promjenama, to može predstavljati problem za rastuće potrebe organizacije.

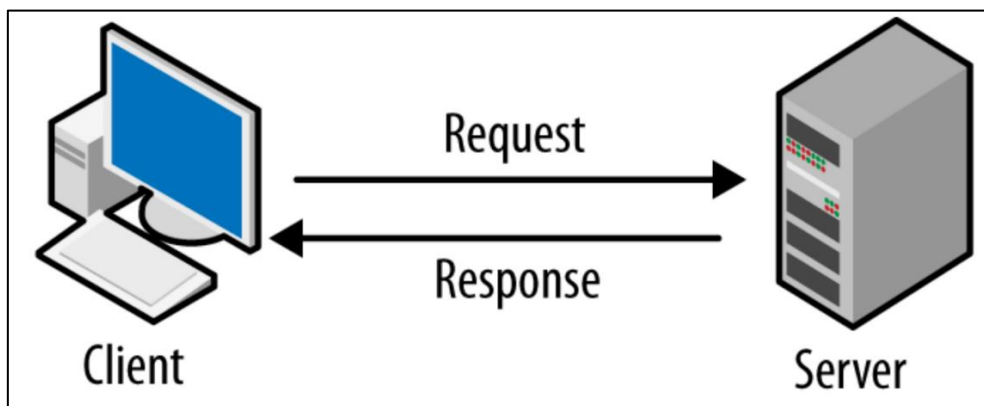
6.2. Dizajn informacijskog sustava

Nakon prikupljenih zahtjeva specifikacije se pretvaraju u izvršni sustav. Prvo se definira hardver i alati u kojima će se izraditi informacijski sustav. Budući da je suvremeno okruženje u poslovanju nezamislivo bez interneta točnije web aplikacija odlučeno je da će informacijski sustav biti izrađen u obliku web aplikacije.

6.3. Arhitektura

S tim u vidu od alata je odabran programski jezik Java točnije njegova najsvremenija inačica za izradu web aplikacija Java Spring Boot². Ova inačica podržava MVC3 (Model, View, Controller) princip projektiranja aplikacija pri čemu su sve komponente zasebne te se lagano mogu mijenjati bez da utječu na ostatak aplikacije.

Slika 3. Prikaz Client-Server modela



Izvor: https://darvishdarab.github.io/cs421_f20/docs/readings/client_server/ (pristupio: 3.12.2023.)

Arhitektura aplikacije pokazuje kako se njeni dijelovi slažu na visokoj razini i za izradu završnog rada koristio se klijent-poslužitelj oblik arhitekture aplikacije. Isti program će prikazivati korisničko sučelje, pristupati podacima i obrađivati osnovne CRUD⁴ (Create, Read, Update i Delete) funkcije. Ova arhitektura koristi bazu podataka koja je smještena na poslužitelju (engl. server) dok računalo korisnika na kojemu je pokrenuta aplikacija služi za pristup tim podacima sa servera i naziva se klijent (engl. client)⁵. Ovdje je omogućeno da više klijentskih upita za podacima bude obrađeno istovremeno i klijenti tj. korisnici dobivaju informacije iz baze istovremeno.

² Java Spring Boot je alat napravljen od strane korisnika koji čini razvoj web aplikacija i mikroservisa sa Spring Frameworkom bržim i lakšim.

³ MVC je obrazac u dizajnu softvera koji se obično koristi za implementaciju korisničkih sučelja, podataka i logike upravljanja, a karakterizira ga razdvajanje poslovne logike softvera i zaslona.

⁴ CRUD je akronim koji dolazi iz svijeta računalnog programiranja i odnosi se na četiri funkcije koje se smatraju potrebnima za implementaciju trajne aplikacije za pohranu: stvaranje, čitanje, ažuriranje i brisanje.

⁵ Model klijent-poslužitelj je dizajn mrežnog računalnog sustava koji ilustrira odnos između dva ili više računala, gdje klijentska računala traže i primaju usluge ili resurse od moćnog centraliziranog poslužiteljskog računala.

6.4. Korisničko sučelje

Nakon određivanja arhitekture aplikacije razmatrao se dizajn korisničkog sučelja. Korisničko sučelje se skiciralo i zatim obradilo u vizualnom dijelu programa Java Spring Boot. Glavne metode za navigaciju su u sklopu CRUD arhitekture, znači korisniku su preko komandi ponuđene funkcije za unos podataka, prikaz podataka, izvršenje kalkulacije koja se traži, promjenu podataka i na kraju brisanje podataka ako je potrebno iz baze.

Uz osnovni stil navigacije aplikacije, dizajn korisničkog sučelja opisuje sve posebne značajke, kao što su ikonice na koje se može kliknuti, važne tablice i metode za određivanje postavki sustava. Ovaj dio dizajna je riješio opća pitanja poput izgleda u boji, fonta i razmještaja ikona i polja. Dizajn više razine je odredio glavne vrste klasa koje će aplikacija koristiti. U nižoj razini dizajna aplikacije definirane su posebne klase koje trebaju programu poput onih koje sadržavaju definicije materijala i metode koje se izvršavaju nad sastavnicama proizvodnje.

6.5. Baza podataka

Dizajn baze podataka izrazito je važan dio aplikacije. U aplikaciji za izračun proizvodnih cijena potrebna je relacijska baza podataka te se koristi jednostavna MySQL baza podataka pisana u SQL6 (engl. Structured Query Language) jeziku koji je danas jedan od industrijskih standarda za kreiranje baza podataka. Pošto se koristi relacijska baza podataka, tijekom dizajniranja više razine skicirale su se tablice koje će sadržavati elemente poput sastavnica proizvoda, cijena i njihove odnose.

Svaka tablica je razrađena sa svojim specifičnim poljima npr. tablica materijala s poljima gdje su navedene karakteristike materijala, količina i ostali relevantni podaci. Svaka tablica ima određeni primarni ključ (engl. Primary key⁷) pomoću kojega je povezana s ostalim tablicama u relacijskom modelu baze kakva je MySQL⁸ baza.

⁶ Structured Query Language (SQL) je standardizirani programski jezik koji se koristi za upravljanje relacijskim bazama podataka i izvođenje različitih operacija nad podacima u njima.

⁷ Primary key je stupac ili stupci koji sadrže vrijednosti koje jedinstveno identificiraju svaki red u tablici.

⁸ MySQL je sustav za upravljanje relacijskim bazama podataka otvorenog koda.

6.6. Testiranje

Pri kraju izrade aplikacije provodi se testiranje rada same aplikacije. Testiranje služi da se provjeri ponašanje sustava u određenim okolnostima i s određenim ulaznim podacima. Prilikom testiranja se provjerava radi li program ono što bi trebao raditi i za što je napravljen. Prilikom testiranja informacijskog sustava za izračun proizvodnih cijena provodilo se jedinično testiranje. Prilikom izrade programskog koda svaka komponenta aplikacije se pojedinačno testirala na greške i one su ispravljane u ranoj fazi kad ih je jednostavnije ispraviti.

7. IZGRADNJA INFORMACIJSKOG SUSTAVA

Informacijski sustav radi svoje sigurnosti mora imati učinkovit sustav za prijavu. Ovaj informacijski sustav ima sučelje za prijavu koje se sastoji od dijela za registraciju korisnika i dijela za autorizaciju korisnika. U sučelju su također uvrštene validacije kao na primjer provjera lozinke i ispravnost unesenih podataka. Ove funkcionalnosti su dio paketa *WebSecurity* koji je sastavni dio Java Spring razvojnog okruženja. Osim toga, u aplikaciji je prisutna funkcionalnost CRUD (Create, Read, Update, Delete) za upravljanje podacima i ima ključnu ulogu u održavanju integriteta i aktualnosti informacija unutar sustava. CRUD, engleska kratica za "Create" (stvaranje), "Read" (čitavanje), "Update" (ažuriranje) i "Delete" (brisanje), predstavlja glavni princip upravljanja podacima unutar informacijskog sustava. Projekt je koncipiran na način da ima dva osnovna segmenta: prvi je prijava korisnika s određenim privilegijama, dok je drugi segment pregled rezultata kalkulacija nakon izvođenja određenih CRUD operacija, ovisno o odobrenim privilegijama.

7.1. Programski alati

Programski alati koji se koriste u radu su trenutno najaktualniji alati koje koristi velika većina developera u svijetu te se koriste njihove najstabilnije verzije.

7.1.1. Eclipse

Za samo pisanje koda koristi se Eclipse editor. Eclipse editor je popularan integrirani razvojni okoliš (IDE) koji se koristi za razvoj različitih vrsta softverskih aplikacija, uključujući Java aplikacije. Eclipse dolazi s integriranim tekstualnim uređivačem koji omogućuje programerima da uređuju izvorni kod svojih projekata. Eclipse Editor nudi niz korisnih značajki za programiranje, kao što su: sintaksno označavanje, automatsko završavanje koda, integracija s debuggerom i upravljanje projektima. Eclipse Editor omogućuje organizaciju projekata, što je ključno za razvoj sofisticiranih aplikacija. Također se u okviru editora može instalirati različite dodatke (plugins) kako bi se proširila funkcionalnost Eclipse Editora i prilagodili ga specifičnim potrebama.

7.1.1. MySQL Workbench

Za bazu podataka koristi se MySQL Workbench alat. MySQL Workbench je vizualni integrirani razvojni okoliš (IDE) koji se koristi za upravljanje MySQL bazama podataka i razvoj aplikacija koje koriste MySQL kao svoj sustav za upravljanje bazama podataka. Ovaj alat omogućuje programerima, DBA-ima (Database Administrators) i inženjerima baza podataka da lako projektiraju, razvijaju i upravljaju MySQL bazama podataka. Ključne značajke MySQL da je omogućeno vizualno projektiranje strukture baze podataka pomoću dijagrama entiteta i veza (ER dijagrami). Mogu se definirati tablice, ključeve, poveznice i drugi elemente koristeći grafičko sučelje. Također je integriran SQL uređivač koji omogućuje pisanje i izvršavanje SQL upita izravno u alatu. Pruža naglašavanje sintakse, automatsko završavanje koda i druge korisne značajke. MySQL Workbench je besplatan alat razvijen od strane MySQL-a i čini razvoj i upravljanje MySQL bazama podataka učinkovitijim i jednostavnijim.

7.2. Dizajn sučelja

Dizajn sučelja izrađen je uz pomoć Thymeleaf okvira za izradu sučelja i Bootstrap okvira za uljepšavanje izgleda sučelja. Thymeleaf je popularan Java templating engine (sustav gotovih predložaka) koji se često koristi za izgradnju dinamičkih web aplikacija. Ovaj alat omogućuje programerima da stvaraju HTML, XML, JavaScript, CSS i druge predloške koji se mogu lako integrirati u web aplikacije. Thymeleaf omogućuje programerima da lako integriraju dinamičke podatke iz Java aplikacija u svoje predloške. To se postiže putem izraza koji se pišu unutar HTML oznaka.

7.2.1. Bootstrap

Bootstrap je besplatan i open-source CSS (Cascading Style Sheets) okvir za izradu modernih i responzivnih web stranica i web aplikacija. Ovaj okvir pruža kolekciju gotovih stilova, komponenata i JavaScript elemenata koji olakšavaju dizajniranje atraktivnih i funkcionalnih web sučelja. Responzivne web stranice su one koje se automatski prilagođavaju različitim veličinama zaslona, uključujući računalne monitore, tablete i pametne telefone. Ovo je ključno za osiguranje da web stranica dobro izgleda na svim uređajima. Bootstrap dolazi s mnogim unaprijed definiranim stilovima za tipografiju, boje, gumbi, forme, navigaciju, kartice, modale i mnoge druge komponente.

7.3. Autorizacija

Prilikom prijavljivanja postoje tri razine privilegija za korisnike: administrator, user ili korisnik i supervizor ili menadžer.

Administrator ima najviše ovlasti i može pregledavati sve tablice, dodavati cijene sastavnicama od kojih nastaju proizvodi ili cijenu sata rada stroja ili radnika. Također on ima pravo dodavati nove korisnike i dodjeljivati im privilegije.

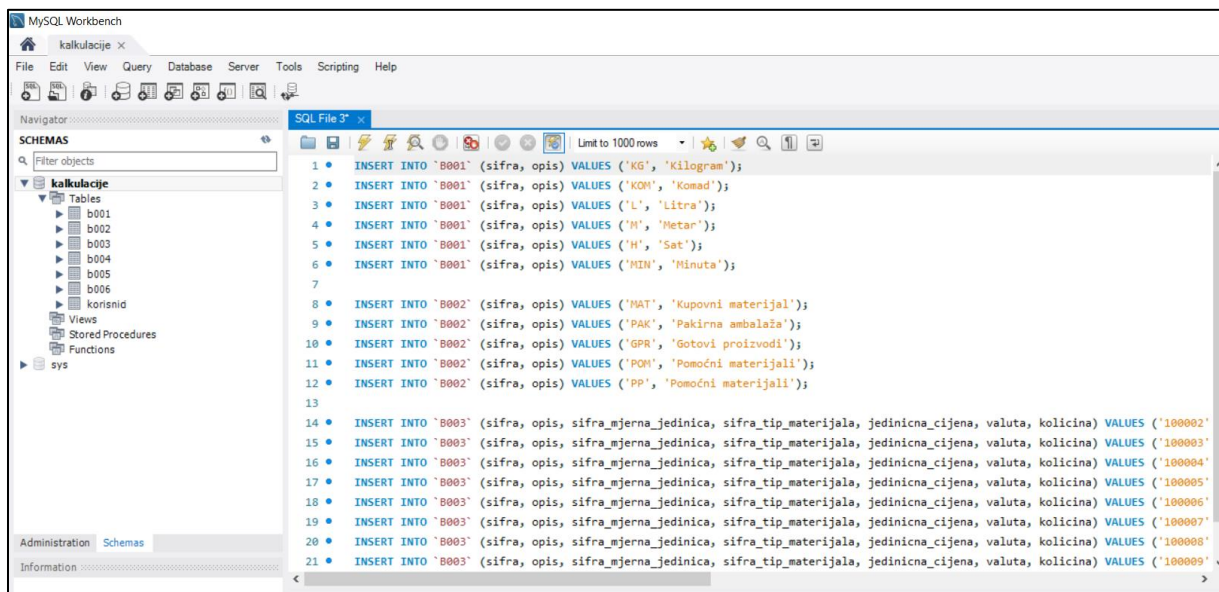
User ili korisnik, nakon uspješne prijave, može uređivati podatke (dodavati količinu proizvoda ili poluproizvoda) te pregledavati rezultat kalkulacija. User nema ovlasti za stvaranje novih materijala ili korisnika te ne može brisati podatke.

Supervizor ili menadžer ima pristup samo čitanju rezultata kalkulacija. U svakom informacijskom sustavu, sigurnost je od izuzetne važnosti. U tu svrhu, koristi se kontrola pristupa temeljena na ulogama kako bi se osiguralo da svaki korisnik ima pristup samo onim podacima i funkcijama koje su relevantne za njegovu ulogu. Korisničko ime, u ovome slučaju je to email korisnika i lozinka su mehanizmi koji se koriste za provjeru i potvrdu autentičnosti korisnika tijekom procesa prijave.

7.4. Dizajn baze podataka

Uz dizajn baze podataka usko su vezani entiteti u Modelu. Baza je dizajnirana po principu da pokriva sve matične podatke koji su se prije čuvali u Excel tablicama. Svaka Excel tablica pretvorena je u tablicu u bazi podataka i ona je ujedno klasa u Modelu aplikacije. Svaka tablica ima svoj posebni *id* koji je ujedno *Primary key*. Pomoću njega tablica se spaja s ostalim tablicama i pomoću njega se na tablici vrše CRUD operacije pomoću SQL upita. Tablice su usko vezane s klasama *repository* za svaki *model*. U *repository* klasama se pišu funkcije koje su usko specijalizirane za dohvaćanje pojedinih podataka koji su potrebni u daljnjoj obradi.

Slika 4. Prikaz baze podataka u MySQL Workbench sučelju



Izvor: Autor

7.4.1. Tablice

Tablice u bazi se podudaraju s tablicama u Excelu. To su tablica korisnici, tablica materijali, tablica mjerna jedinica, tablica sastavnice, tablica operacije, tablica strojevi i tablica tip materijala.

Tablica korisnici u sebi sadrži podatke o id-ju (to je primarni ključ tablice), email, lozinka, enabled, status i uloga.

Tablica materijali u sebi sadrži podatke o šifri materijala (to je primarni ključ tablice), jedinična cijena, količina, opis, valuta te je povezana sa tablicama mjerna jedinica i tip materijala.

Tablica mjerna jedinica u sebi sadrži podatke o šifri mjerne jedinice (to je primarni ključ tablice) te opis.

Tablica strojevi u sebi sadrži podatke o šifri stroja (to je primarni ključ tablice), cijena rada radnika, cijena rada stroja, opis, valuta, sata rada radnika potrebni za proizvodnju te sate rada stroja potrebnih za proizvodnju određene količine proizvoda i povezana je s tablicom mjerna jedinica.

Tablica tip materijala u sebi sadrži podatke o šifri tipa materijala (to je primarni ključ tablice) i opis.

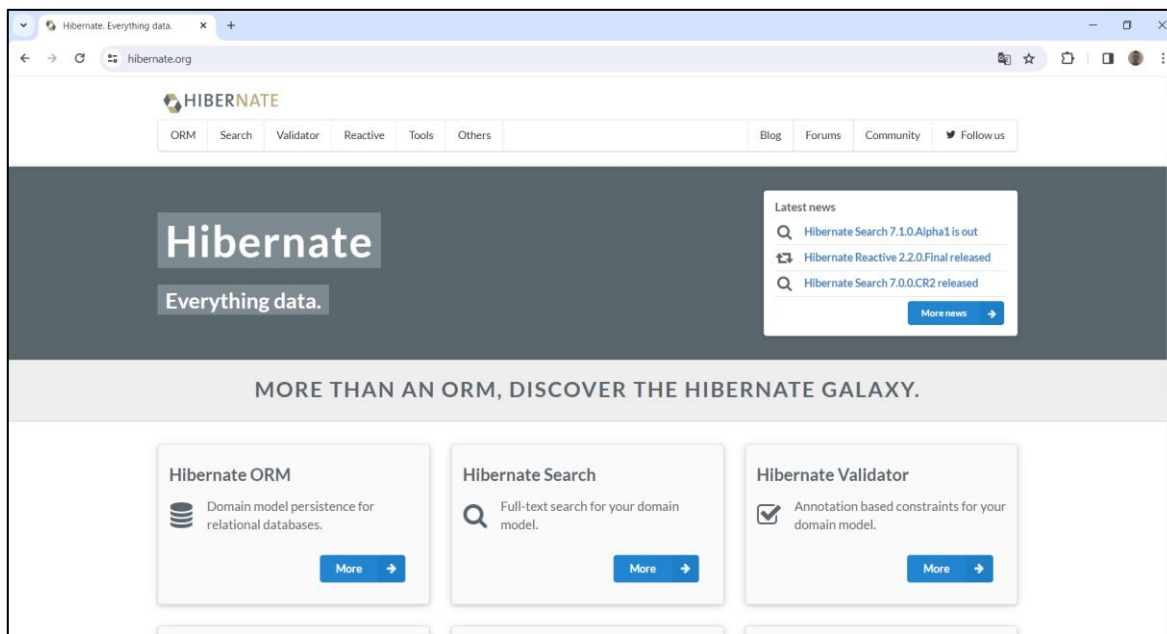
Tablica operacije koja sadrži normative vremena ima primarni ključ u broju normativa i sadrži još šifru materijala, opis materijala, mjernu jedinicu pojedinog materijala, količinu materijala, šifru stroja potrebnog za tu operaciju, opis stroja za određenu operaciju, vrijeme stroja potrebno za operaciju i vrijeme radnika potrebno za neku operaciju određeno u satima.

Tablica sastavnice u sebi sadrži podatke o šifri i broju normativa pojedine sastavnice (to je primarni ključ tablice koji je sastavljen od dva primarna ključa po kojima se može točnije pronaći ta sastavnica te se takav primarni ključ naziva kompozitni primarni ključ predstavljen u klasi *SastavnicePK*), količina, opis te je povezana sa tablicama stroj i mjerna jedinica. Radi brzine i jednostavnosti se koristi Hibernate ORM koji mapira podatke iz modela u podatke u tablicu u bazi.

7.4.1. Hibernate

Hibernate je popularni Java okvir za objektno-relacijsko mapiranje (ORM), što znači da omogućava programerima da rade s objektima u programskom jeziku Java umjesto da se bave SQL upitima za interakciju s bazama podataka. Hibernate pojednostavljuje razvoj aplikacija koje koriste baze podataka tako što omogućava mapiranje Java objekata na tablice u bazi podataka i obrnuto. Ključna značajka mu je da omogućava programerima definiranje Java objekata koji odražavaju strukturu podataka u bazi podataka. Tako se Java objekti automatski povezuju s odgovarajućim tablicama u bazi podataka. Hibernate se često koristi u Java aplikacijama koje koriste baze podataka za pohranu i upravljanje podacima. Omogućava programerima da se usmjere na logiku aplikacije umjesto da se bave složenostima interakcije s bazama podataka.

Slika 5. Prikaz Hibernate grafičkog sučelja

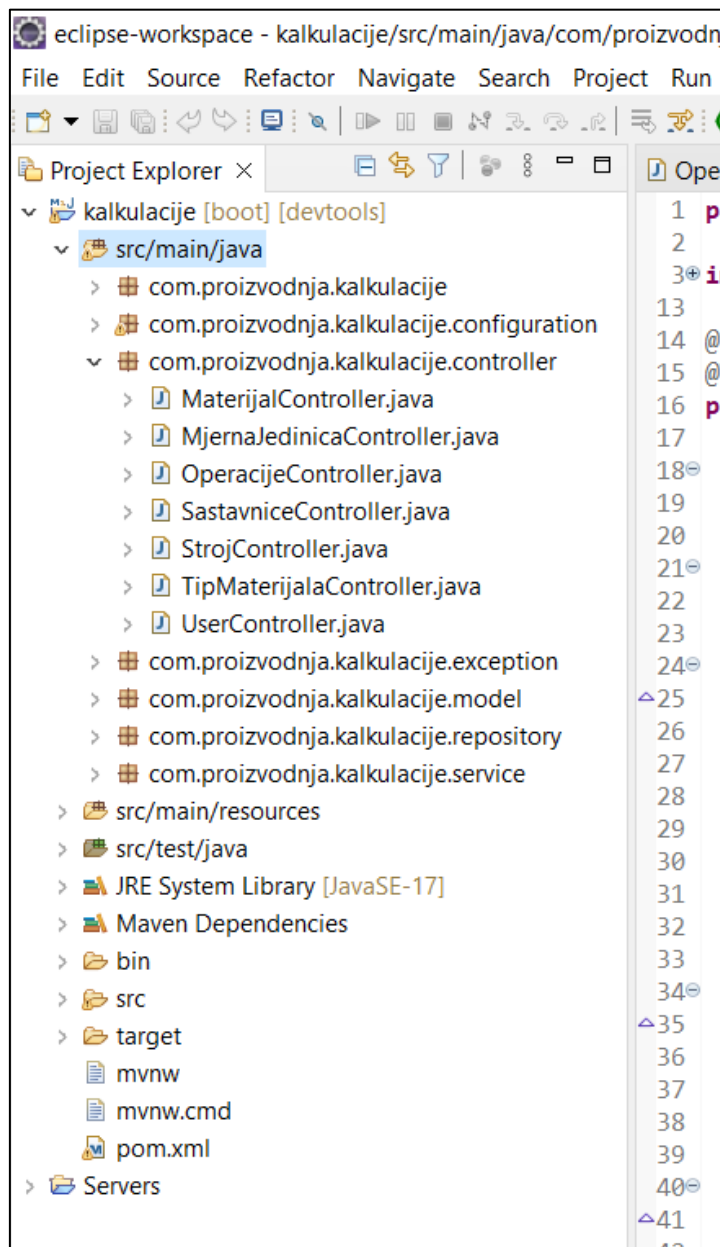


Izvor: <https://hibernate.org/> (pristup: 2.12.2023.)

8. DIJELOVI APLIKACIJE

Aplikacija je podijeljena na razne modularne elemente radi lakšeg održavanja po principu koji je ustaljen prilikom izrade web aplikacija u Java Spring programskom okviru. Kod je podijeljen na više slojeva.

Slika 6. Prikaz arhitekture aplikacije u Eclipse IDE



Izvor: Autor

Slika 7. Prikaz pokretanja Spring aplikacije u Eclipse IDE

```

kalkulacije - KalkulacijeApplication [Spring Boot App] C:\Program Files\Java\jdk-17\bin\javaw.exe (2. pro 2023. 17:17:59) [pid: 8764]
2023-12-02T17:18:04.713+01:00 INFO 8764 --- [ restartedMain] org.hibernate.cfg.Environment : HHH000406: Using bytecode reflection optimizer
2023-12-02T17:18:04.959+01:00 INFO 8764 --- [ restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name : bytebuddy
2023-12-02T17:18:05.211+01:00 INFO 8764 --- [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2023-12-02T17:18:05.254+01:00 INFO 8764 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-12-02T17:18:05.695+01:00 INFO 8764 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jdbc.Connection
2023-12-02T17:18:05.700+01:00 INFO 8764 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-12-02T17:18:06.183+01:00 INFO 8764 --- [ restartedMain] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name : bytebuddy
2023-12-02T17:18:07.117+01:00 INFO 8764 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.
2023-12-02T17:18:07.416+01:00 INFO 8764 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'c
2023-12-02T17:18:07.915+01:00 INFO 8764 --- [ restartedMain] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If applicable, HQL parser will be
2023-12-02T17:18:08.574+01:00 WARN 8764 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, da
2023-12-02T17:18:08.909+01:00 INFO 8764 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.
2023-12-02T17:18:09.170+01:00 WARN 8764 --- [ restartedMain] .s.s.UserDetailsServiceAutoConfiguration :

Using generated security password: c91e6d21-b80a-48e7-9f76-914b40f04cf3

This generated password is for development use only. Your security configuration must be updated before running your application in production.

2023-12-02T17:18:09.236+01:00 INFO 8764 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-12-02T17:18:09.293+01:00 INFO 8764 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9045 (http) with context path ''
2023-12-02T17:18:09.312+01:00 INFO 8764 --- [ restartedMain] c.p.kalkulacije.KalkulacijeApplication : Started KalkulacijeApplication in 8.115 seconds (process run
2023-12-02T17:18:59.844+01:00 INFO 8764 --- [nio-9045-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-12-02T17:18:59.845+01:00 INFO 8764 --- [nio-9045-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-12-02T17:18:59.847+01:00 INFO 8764 --- [nio-9045-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms

```

Izvor: Autor

8.1. Model

Prva i glavna mapa je model. U njoj se nalaze klase koje su istovjetne tabelama u bazi podataka. Model je povezan sa servisnom mapom u kojoj su nacrti funkcija koje ima pojedina klasa u modelu. To je sve stavljeno u klase pod nazivom *interface* i svaka interface klasa ima svoju implementaciju. U aplikaciji su prisutni modeli klasa Materijal, TipMaterijala, MjernaJedinica, Stroj, Sastavnice, Operacije, SastavnicePK te User. Klase su istovjetne tablicama u bazi podataka.

8.2. Service

U okviru *Spring Frameworka*, "service" je pojam koji se često koristi za označavanje sloja poslovne logike ili servisa u aplikaciji. Ovaj sloj obično sadrži poslovnu logiku koja implementira određenu funkcionalnost ili operaciju koja se izvršava unutar aplikacije.

Osnovna ideja Spring servisa je da se poslovna logika izdvoji iz kontrolera (web sloj) i repozitorija (sloj podataka) te smjesti u odvojeni sloj. To poboljšava modularnost, čitljivost i održivost koda. Servisi često obavljaju operacije poput obrade poslovnih pravila, komunikacije s repozitorijima i integracije s vanjskim sustavima.

U Springu se servisi obično označavaju anotacijom `@Service`. Ova anotacija omogućuje Spring kontejneru da prepozna taj razred kao servis i automatski ga upravlja, uključujući i ubrizgavanje ovisnosti (*Dependency Injection*). U servisnoj implementaciji su funkcije koje su planirane u *interface* klasi razrađene u detalje i postignuta je njihova funkcionalnost. Izračuni troškova materijala temelje se na količinama i cijenama jedinica komponenti koje se dohvaćaju iz repozitorija *MaterijalRepository* i *StrojRepository*.

Kôd 1. Prikaz rekurzivne funkcije za kalkulaciju normativa materijala

```
@Override
public Float kalkulacijaNormativMaterijala(String brojNormativa, Integer kolicina) {
    return kalkulacijaTrosakMaterijala(brojNormativa, kolicina);
}
private Float kalkulacijaTrosakMaterijala(String brojNormativa, Integer kolicina) {
    Optional<List<Sastavnice>> podsastavnice = repository.findSastavniceByBrojNormativa(brojNormativa);
    if (podsastavnice.isPresent()) {
        List<Sastavnice> stavke = podsastavnice.get();
        Integer unesenaKolicinaZaglavlja = repository.findZaglavljeByBrojNormativa(brojNormativa).get().getKolicina();
        Float omjer = (float) (kolicina) / unesenaKolicinaZaglavlja;
        Float ukupnaCijenaMaterijala = 0.00f;
        Float cijenaMaterijala = 0.00f;
        Float jedinicaCijena = 0.00f;
        for (int i = 0; i < stavke.size(); i++) {
            Integer kolicinaStavke = stavke.get(i).getKolicina();

            if (stavke.get(i).podsastavniceStavke() {
                Optional<Sastavnice> podzaglavlje = repository.findZaglavljeBySifraMaterijala(stavke.get(i).getId().getSifra());
                String brojNormativaPodzaglavlja = podzaglavlje.get().getId().getBrojNormativa();
                ukupnaCijenaMaterijala += kalkulacijaTrosakMaterijala(brojNormativaPodzaglavlja, kolicina);
            }
            else {
                jedinicaCijena = (repositoryMaterijal.findById(stavke.get(i).getId().getSifra()).get().getJedinicaCijena());
                cijenaMaterijala = kolicinaStavke * omjer * jedinicaCijena;
                ukupnaCijenaMaterijala += cijenaMaterijala;
            }
        }
        return ukupnaCijenaMaterijala;
    }
    else {
        return null;
    }
}
```

Izvor: Autor

8.3. Kontroler

Sljedeći sloj je sloj kontrolera. U tom sloju su funkcije koje se nadovezuju na servisni sloj i prenose funkcionalnosti sa servisnog sloja na MVC (*model-view-controller*) koji prikazuje funkcionalnosti na grafičkom sučelju. Tu se koristi *Thymeleaf* koji je povezan sa HTML-om (*Hyper Text Markup Language*) i CSS-om (*Cascade Style Sheet*). Pomoću ovih funkcionalnosti podaci se prikazuju korisniku na web stranicama određenog web sjedišta. U kontroler se upisuju *url* adrese po kojima onda web preglednik preusmjerava korisnika na određenje dijelove aplikacijskog sučelja.

Kôd 2. Prikaz funkcija u klasi kontroler

```
@GetMapping("/uredi_materijal/{sifra}")
public ModelAndView urediMaterijal(@PathVariable("sifra") String sifra) {
    ModelAndView mav = new ModelAndView("uredi_materijal");
    Materijal materijal = service.getMaterijal(sifra);
    mav.addObject("materijal", materijal);
    return mav;
}

@PostMapping("/uredi_materijal")
public String spremiMaterijal(@ModelAttribute("materijal") Materijal materijal) {
    service.updateMaterijal(materijal);
    return "redirect:/materijali/";
}
```

Izvor: Autor

8.4. Repozitorij

Sloj repozitorija služi da se u njega upisuju posebne funkcije koje imaju u sebi direktne upite u SQL jeziku kojima se podacima manipulira direktno u bazi podataka. Ti upiti nazivaju se *queries* i postoji posebna oznaka za takve funkcije koje u sebi imaju upite, iznad funkcije se stavlja anotacija *@Query*.

Kôd 3. Prikaz repository funkcija za dohvat sastavnica iz baze podataka

```
@Repository
public interface SastavniceRepository extends JpaRepository<Sastavnice, SastavnicePK>{

    Optional<Sastavnice> findById(SastavnicePK id);

    @Query("SELECT s FROM Sastavnice s WHERE s.id.brojNormativa = :brojNormativa AND s.stavka_zaglavlje LIKE 'S%'")
    Optional<List<Sastavnice>> findSastavniceByBrojNormativa(@Param("brojNormativa") String brojNormativa);

    @Query("SELECT s FROM Sastavnice s WHERE s.id.brojNormativa = :brojNormativa AND s.stavka_zaglavlje LIKE 'Z%'")
    Optional<Sastavnice> findZaglavljeByBrojNormativa(@Param("brojNormativa") String brojNormativa);

    @Query("SELECT z FROM Sastavnice z WHERE z.stavka_zaglavlje LIKE 'Z%'")
    List<Sastavnice> findZaglavlja();

    @Query("SELECT z FROM Sastavnice z WHERE z.id.sifra = :sifra AND z.stavka_zaglavlje LIKE 'Z%'")
    Optional<Sastavnice> findZaglavljeBySifraMaterijala(@Param("sifra") String sifra);
}
```

Izvor: Autor

8.5. Rukovođenje greškama i iznimkama

U posebnom modulu *exception* su obrađeni krajnji putevi za ovladavanje greškama. Ukoliko se pojavi greška tu su definirane funkcije koje tu grešku ili preusmjere ili obavijeste korisnika da se javila greška.

8.6. Konfiguracija

Konfiguracijski modul ili *configuration* služi da bi se u aplikaciju implementirale funkcije koje su vezane uz sigurnost na internetu. One korisniku omogućuju stvaranje sesije koja se pokreće autentifikacijom i autorizacijom. Prije njih je prisutna validacija po emailu i lozinki. Za autentifikaciju je zadužena klasa *AuthenticationHandler* sa svojim funkcijama dok je za autorizaciju zadužena klasa *SecurityConfig* i *WebMvcConfig* sa svojim funkcijama.

Kôd 4. Prikaz konfiguracijske funkcije

```
@Autowired
public void configureGlobal(AuthenticationManagerBuilder auth)
    throws Exception {
    auth.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery("SELECT email, lozinka, enabled "
            + "from korisnici "
            + "where email = ?")
        .authoritiesByUsernameQuery("select email, uloga "
            + "from korisnici "
            + "where email = ?");
}
```

Izvor: Autor

8.7. Statički dio aplikacije

U statičkom dijelu aplikacije nalazi se modul za gotovim web predlošcima za svako pojedino sučelje u HTML formatu. Tu se nalaze predlošci za sučelje prijave (login.html), za sučelje registracije (registracija.html) te sučelja za tri razne uloge koje imaju korisnici (pocetna_admin.html, pocetna_supervisor.html, pocetna_user.html). U *application.properties* modulu nalaze se sve važne informacije za aplikaciju. Tu se url kojim je aplikacija povezana sa bazom. Ovdje se također definira port na kojemu program *Apache Tomcat* stvara lokaliziranu web stranicu za prikaz rezultata aplikacije.

Slika 8. Prikaz statičkog dijela aplikacije s upotrebom Thymeleaf-a

```
52 </nav>
53< <div class="container table-container mx-auto text-center">
54< <table class="table table-striped">
55< <thead class="thead-light">
56< <tr>
57< <th>Šifra</th>
58< <th>Opis</th>
59< <th>Tip materijala</th>
60< <th>Cijena</th>
61< <th>Valuta</th>
62< <th>Količina</th>
63< <th>Mjerna jedinica</th>
64< <th>Akcija</th>
65< </tr>
66< </thead>
67< <tbody>
68< <tr th:each="materijal : ${materijali}">
69< <td th:text="${materijal.sifra}"></td>
70< <td th:text="${materijal.opis}"></td>
71< <td th:text="${materijal.tipMaterijala}"></td>
72< <td th:text="${materijal.jedinicnaCijena}"></td>
73< <td th:text="${materijal.valuta}"></td>
74< <td th:text="${materijal.kolicina}"></td>
75< <td th:text="${materijal.mjernaJedinica}"></td>
76< <td>
77< <a th:href="@{/uredi_materijal/{sifra}(sifra=${materijal.sifra})}"
78< class="btn btn-warning"
79< th:unless="${materijal.tipMaterijala.sifra == 'GPR' || materijal.tipMaterijala.sifra == 'PP'}">
80< Uredi</a>
81< </td>
82< </tr>
83< </tbody>
84< </table>
85< </div>
86< <footer class="text-center text fixed-bottom" style="background-color: #00695c;">
87< <div class="container p-4">
88< <a class="text-white">KALKULACIJE U PROIZVODNJI</a>
89< </div>
90< <div class="text-center p-3" style="background-color: white;">
```

Izvor: Autor

8.8. Apache Maven

U aplikaciji koristim *Apache Maven* alat. Apache Maven je popularni alat za upravljanje projektima i izgradnju softvera. Korišten je u razvoju Java aplikacija, ali se može prilagoditi za različite programske jezike i okoline. Maven omogućuje programerima da definiraju ovisnosti (biblioteke i komponente koje aplikacija koristi) u jednostavnom XML formatu. Maven automatski preuzima ovisnosti iz centralnog repozitorija i upravlja njihovim verzijama. Maven također automatizira proces izgradnje projekta. To uključuje kompilaciju izvornog koda, upravljanje resursima, izradu arhive (JAR), i druge zadatke. Koristi se plugin sustav koji omogućava programerima da prošire njegove mogućnosti. Postoje nastavci (*plugins*) za različite zadatke, uključujući izgradnju, upravljanje ovisnostima, i izvođenje testova. Na kraju se formira POM (*Project Object Model*) datoteka za svaki projekt (pom.xml) koja sadrži konfiguraciju projekta, ovisnosti i druge informacije o projektu.

Kôd 5. Prikaz umetanja Maven ovisnosti u aplikaciju

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Izvor: Autor

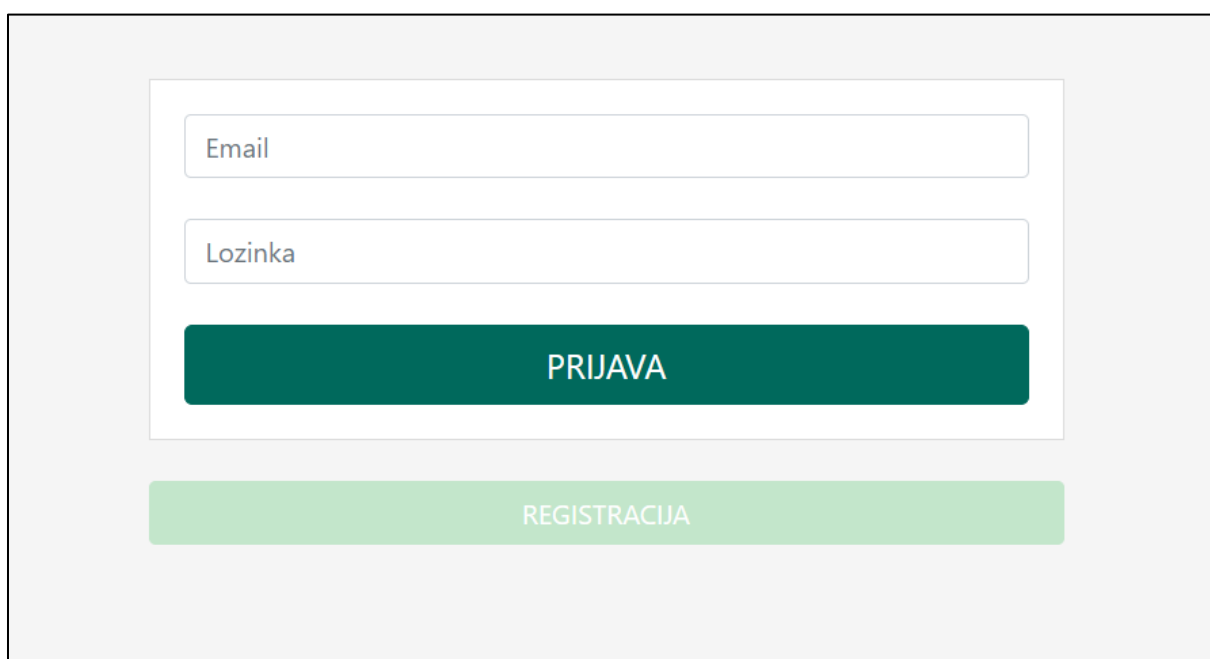
8.9. Apache Tomcat

Za potrebe izvođenja aplikacije na portu lokalnog poslužitelja koristio sam Apache Tomcat alat. Apache Tomcat je otvoren izvor (open-source) servlet kontejner koji omogućuje izvođenje Java Servleta, JavaServer Pages (JSP) i drugih Java web aplikacija. Tomcat je razvijen od strane Apache Software Foundation i jedan je od najpopularnijih web servera za izvođenje Java web aplikacija. Tomcat uključuje ugrađeni HTTP server koji omogućuje obradu HTTP zahtjeva i odgovora.

9. PRIKAZ SUČELJA I FUNKCIONALNOSTI

Ulazak u sustav je omogućen prijavom u korisničko sučelje. Ukoliko korisnik nije registriran on se registrira pod imenom i email-om te odabere lozinku. Nakon što je korisnik kreiran ovisno o završetku email-a (npr. @user.com) daje mu se uloga u sustavu i razina prava. Ovdje je prikazano ulazno korisničko sučelje za prijavu korisnika.

Slika 9. Prikaz login forme za ulazak u program



The image shows a login form with two input fields: 'Email' and 'Lozinka'. Below the fields is a dark green button labeled 'PRIJAVA'. Below the button is a light green button labeled 'REGISTRACIJA'. The form is centered on a light gray background.

Izvor: Autor

Prilikom odabira tipke materijali pokazuje se tablica s materijalima iz baze podataka sa stupcima među kojima su podaci o šifri materijala, opisu, tipu materijala, valuti materijala, mjernoj jedinici. Na kraju kolone nalazi se gumbi za uređivanje materijala. Pritiskom na gumb javlja se sučelje u kojemu korisnik može unijeti cijenu materijala i jediničnu količinu.

Slika 10. Prikaz tablice materijala

Šifra	Opis	Tip materijala	Cijena	Valuta	Količina	Mjerna jedinica	Akcija
100002	Strech folija	Pakirna ambalaža	3.6	EUR	1	Kilogram	Uredi
100003	Ljepilo	Pomoćni materijali	1.0	EUR	1	Kilogram	Uredi
100004	Naljepnica	Pakirna ambalaža	2.0	EUR	1	Komad	Uredi
100005	Omot	Pakirna ambalaža	3.0	EUR	1	Komad	Uredi
100006	Šećer	Kupovni materijal	4.0	EUR	1	Kilogram	Uredi
100007	Aroma 1	Kupovni materijal	6.0	EUR	1	Kilogram	Uredi
100008	Lecitin	Kupovni materijal	1.0	EUR	1	Kilogram	Uredi
100009	Mlijeko	Kupovni materijal	2.0	EUR	1	Litra	Uredi

KALKULACIJE U PROIZVODNJI

Izvor: Autor

Tablica s tipovima materijala pokazuje koji sve tipovi materijala postoje te u stupcu opis navodi stvarni naziv tipa materijala. Sa strane se nalaze akcijski gumbi pomoću kojih se uređuje svaki pojedini materijal.

Slika 11. Prikaz tablice s tipovima materijala

Šifra	Opis	Akcija
GPR	Gotovi proizvodi	Uredi
MAT	Kupovni materijal	Uredi
PAK	Pakirna ambalaža	Uredi
POM	Pomoćni materijali	Uredi
PP	Pomoćni materijali	Uredi

KALKULACIJE U PROIZVODNJI

Izvor: Autor

Ovdje su prikazani tipovi materijala po svojoj šifri i opisu materijala. Šifre su GPR (Gotovi proizvod), MAT (Kupovni materijal), PAK (Pakirna ambalaža), POM (Pomoćni materijali) i PP (ima oznaku polu-proizvod).

Tablica s mjernim jedinicama pokazuje koji sve tipovi mjernih jedinica postoje te u stupcu opis navodi stvarni naziv mjerne jedinice. Među mjernim jedinicama koje se koriste su sat (H), kilogram (KG), komad (KOM), litra (L), metar (M) i minuta (MIN).

Slika 12. Prikaz tablice s mjernim jedinicama

Šifra	Opis	Akcija
H	Sat	Uredi
KG	Kilogram	Uredi
KOM	Komad	Uredi
L	Litra	Uredi
M	Metar	Uredi
MIN	Minuta	Uredi

KALKULACIJE U PROIZVODNJI

Izvor: Autor

Tablica s prikazom proizvodnih linija pokazuje koje sve proizvodne linije postoje te koji strojevi su na kojoj proizvodnoj liniji. U stupcu opis navodi se stvarni naziv stroja, a ne samo njegova šifra s kojom se on povezuje sa određenim sastavnicama. Navedena je još cijena rada stroja po satu u eurima i cijena rada radnika po satu u eurima. Te dvije varijable unosi korisnik.

Slika 13. Prikaz proizvodnih linija

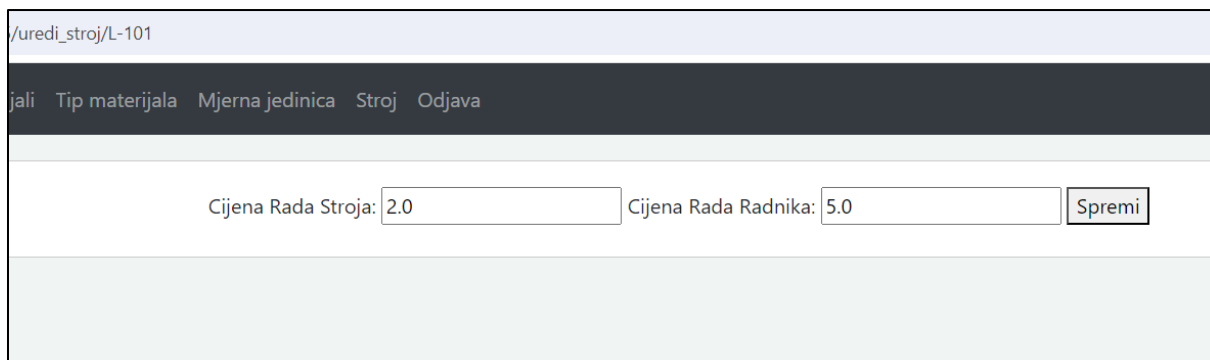
Šifra	Opis	Cijena stroja	Valuta	Cijena radnika	Valuta	Mjerna jedinica	Akcija
L-101	Pakerica čokolade	2.0	EUR	5.0	EUR	Sat	Uredi
L-102	Pakerica bombona	3.0	EUR	3.0	EUR	Sat	Uredi
L-103	Izrada čokolade	2.0	EUR	2.0	EUR	Sat	Uredi
L-104	Izrada čokoladne smjese	3.0	EUR	3.0	EUR	Sat	Uredi
L-105	Dražiranje	4.0	EUR	4.0	EUR	Sat	Uredi
L-106	Izrada bombonske smjese	3.0	EUR	2.0	EUR	Sat	Uredi

KALKULACIJE U PROIZVODNJI

Izvor: Autor

Na slici 14 je prikazano sučelje koje se pojavi nakon što korisnik klikne na gumb uredi pokraj pojedinog stroja. Ovdje korisnik unosi parametre cijene rada radnika i rada stroja po satu prema kojima se kasnije kalkulira cijena proizvodnje.

Slika 14. Prikaz uređivanja proizvodne linije

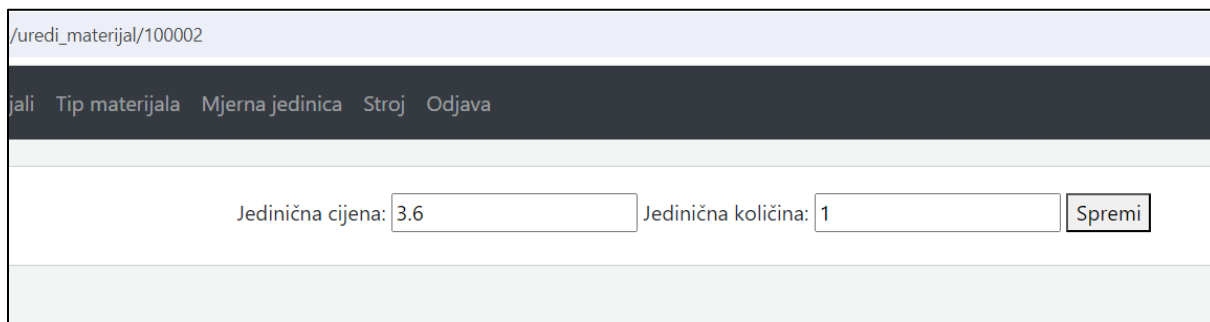


The screenshot shows a web form titled "/uredi_stroj/L-101". At the top, there is a navigation bar with the following items: "jali", "Tip materijala", "Mjerna jedinica", "Stroj", and "Odjava". Below the navigation bar, the form contains two input fields: "Cijena Rada Stroja:" with the value "2.0" and "Cijena Rada Radnika:" with the value "5.0". To the right of these fields is a "Spremi" button. The form is set against a light blue background.

Izvor: Autor

Istovjetno sučelje je prisutno i prilikom unosa cijene materijala za pojedine materijale ukoliko korisnik pritisne gumb uredi, program ga dalje vodi na url `/uredi_materijal` i tu se onda pomoću *GET* i *POST* metoda svakom materijalu upisuje njegova cijena po njegovoj šifri.

Slika 15. Prikaz uređivanja cijene materijala



The screenshot shows a web form titled "/uredi_materijal/100002". At the top, there is a navigation bar with the following items: "jali", "Tip materijala", "Mjerna jedinica", "Stroj", and "Odjava". Below the navigation bar, the form contains two input fields: "Jedinična cijena:" with the value "3.6" and "Jedinična količina:" with the value "1". To the right of these fields is a "Spremi" button. The form is set against a light blue background.

Izvor: Autor

Na kraju se javlja sučelje u kojemu korisnik odabire pojedinu sastavnicu od ponuđenih sastavnica koje su klasificirane kao zaglavlje te se nakon unosa količine te odabrane sastavnice u ovom slučaju pomoćnog proizvoda ili gotovog proizvoda kalkulira cijena proizvodnje iste.

Slika 16. Prikaz uređivanja proizvodne linije

Izvor: Autor

Rezultat kalkulacije se zatim prikazuje na sučelju */rezultat.html*. Ovdje je prikazan odabrani proizvod ili poluproizvod, količina odabranog proizvoda ili poluproizvoda i na kraju cijena proizvodnje istoga.

Slika 17. Prikaz rezultata kalkulacije

Proizvod	Količina	Cijena proizvodnje
MLIJEČNA ČOKOLADA 100g	1	0.4550067 EUR

Izvor: Autor

10. ZAKLJUČAK

Za izgradnju informacijskog sustava koristio se Java Spring Boot razvojni okvir, a za razvoj baze MySQL Workbench i Eclipse IDE kao razvojno okruženje za pisanje koda. Informacijski sustav ima sučelje za prijavu s dijelom za registraciju korisnika i dijelom za autorizaciju korisnika, u kojem su implementirane provjere poput provjere lozinke i ispravnosti unesenih podataka. Ovaj sustav također podržava CRUD operacije (Create, Read, Update, Delete) za upravljanje podacima. Dizajn baze podataka uključuje tablice koje sadrže podatke o materijalima, cijenama i njihovim odnosima, a svaka tablica ima određeni primarni ključ za povezivanje s drugim tablicama u relacijskom modelu baze podataka. Tijekom razvoja, provodila su se jedinična testiranja kako bi se provjerila ispravnost rada aplikacije i ispravile greške u ranoj fazi. Osim toga, radi sigurnosti implementiran je učinkovit sustav za validaciju korisnika i ostalih unosa.

Prilikom izrade završnog rada shvaća se važnost implementacije informacijskog sustava kalkulacija u proizvodnji u suvremeno poslovno okruženje proizvodnih tvrtki. Neophodan je za učinkovito vođenje i upravljanje poslovnim procesima te omogućava brzi uvid u promjene cijena proizvoda ovisno o promjenama na tržištu. Omogućava menadžerima da precizno definiraju svoje zahtjeve prema dobavljačima, radnicima i kupcima. Također, važno je da se sustav prilagodi specifičnim potrebama korisnika.

Zaključak je da se u budućnosti očekuje napredak u složenosti informacijskih sustava kalkulacija u proizvodnji, integrirajući se s naprednim tehnologijama kao što su umjetna inteligencija i Internet stvari (*IoT*). To će omogućiti organizacijama da u stvarnom vremenu prate i prilagođavaju svoje narudžbe, modificiraju cijenu rada kako bi ostali konkurentni sa cijenom svojeg proizvoda. Daljnjim razvojem tehnologije rasti će i uloga informacijskih sustava koji će postajati sve moćniji alat u rukama menadžmenta te će postati ključni instrument u postizanju profita i ostalih svake ozbiljnije korporacije.

Izjava o autorstvu

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

Bana Josipa Jelačića 22/a, Čakovec

IZJAVA O AUTORSTVU

Završni/diplomski rad isključivo je autorsko djelo studenta te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, internetskih i drugih izvora) bez pravilnog citiranja. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom i nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, _____ David Cecelja _____ (ime i prezime studenta) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog rada pod naslovom

_____ Informacijski sustav izračuna proizvodnih cijena _____

te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:



_____ (vlastoručni potpis)

Literatura

1. Nađ J. (2020.): *Programsko inženjerstvo i informacijski sustavi*, Međimursko veleučilište u Čakovcu, Čakovec
2. Pavlić M. (2011.): *Informacijski sustavi*, Školska knjiga, Zagreb.
3. Walls C. (2015.): *Spring Boot in Action*, Manning, New York
4. Trstenjak B. (2011.): *Programiranje: skripta*, Međimursko veleučilište u Čakovcu, Čakovec
5. Client-Server Model, <https://geeksforgeeks.org>, (Datum pristupa: 5.11.2023.)
6. CRUD logika, <https://sumologic.com>, (Datum pristupa: 5.11.2023.)
7. SQL, <https://w3schools.com>, (Datum pristupa: 28.11.2023.)
8. MySQL, <https://mysql.com>, (Datum pristupa: 28.11.2023.)
9. ERP, <https://dynamics.microsoft.com>, (Datum pristupa: 2.12.2023.)
10. MVC, <https://developer.mozilla.org>, (Datum pristupa: 2.12.2023.)
11. Spring by VMware Tanzu, <https://spring.io>, (Datum pristupa: 2.12.2023.)
12. Java, <https://java.com>, (Datum pristupa: 2.12.2023.)
13. Thymeleaf, <https://thymeleaf.org>, (Datum pristupa: 3.12.2023.)
14. Apache Maven Project, <https://maven.apache.org>, (Datum pristupa: 3.12.2023.)
15. Apache Tomcat, <https://tomcat.apache.org>, (Datum pristupa: 3.12.2023.)
16. Eclipse Foundation, <https://eclipse.org>, (Datum pristupa: 3.12.2023.)
17. Internet of Things, <https://techtarget.com>, (Datum pristupa: 4.12.2023.)
18. Java Server Pages, <https://javatpoint.com>, (Datum pristupa: 4.12.2023.)
19. Hibernate, <https://hibernate.org>, (Datum pristupa: 4.12.2023.)
20. Bootstrap, <https://getbootstrap.com>, (Datum pristupa: 4.12.2023.)
21. Spring Boot Tutorial, <https://www.baeldung.com/spring-boot>, (Datum pristupa: 4.12.2023.)

Popis ilustracija

Slika 1. Princip kreiranja proizvodne cijene	12
Slika 2. Princip prikupljanja zahtjeva	15
Slika 3. Prikaz Client-Server modela	21
Slika 4. Prikaz baze podataka u MySQL Workbench sučelju.....	27
Slika 5. Prikaz Hibernate grafičkog sučelja	29
Slika 6. Prikaz arhitekture aplikacije u Eclipse IDE	30
Slika 7. Prikaz pokretanja Spring aplikacije u Eclipse IDE.....	31
Slika 8. Prikaz statičkog dijela aplikacije s upotrebom Thymeleaf-a	35
Slika 9. Prikaz login forme za ulazak u program	37
Slika 10. Prikaz tablice materijala.....	38
Slika 11. Prikaz tablice s tipovima materijala	39
Slika 12. Prikaz tablice s mjernim jedinicama	40
Slika 13. Prikaz proizvodnih linija.....	41
Slika 14. Prikaz uređivanja proizvodne linije	42
Slika 15. Prikaz uređivanja cijene materijala	42
Slika 16. Prikaz uređivanja proizvodne linije	43
Slika 17. Prikaz rezultata kalkulacije	43

Prilozi

Kôd 1. Prikaz funkcije za kalkulaciju normativa materijala u service klasi	32
Kôd 2. Prikaz funkcija u klasi kontroler.....	33
Kôd 3. Prikaz repository funkcija za dohvat sastavnica iz baze podataka	33
Kôd 4. Prikaz konfiguracijske funkcije.....	34
Kôd 5. Prikaz umetanja Maven ovisnosti u aplikaciju.....	36