

Simulacija upravljačko-izvještajnog informacijskog sustava

Šimičić, Ivan

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:622654>

Rights / Prava: [Attribution-NonCommercial-ShareAlike 3.0 Unported / Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 3.0](#)

Download date / Datum preuzimanja: **2024-04-25**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Ivan Šimičić

**SIMULACIJA UPRAVLJAČKO-
IZVJEŠTAJNOG INFORMACIJSKOG
SUSTAVA**

ZAVRŠNI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ivan Šimičić

Matični broj: 0016124532

Studij: Poslovni sustavi

SIMULACIJA UPRAVLJAČKO-IZVJEŠTAJNOG
INFORMACIJSKOG SUSTAVA

ZAVRŠNI RAD

Mentor:

Izv. prof. dr. sc. Igor Balaban

Varaždin, kolovoz 2019.

Ivan Šimičić

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom radu bavit ću se temom informacijskih sustava te simulacijom upravljačko-izvještajnog informacijskog sustava. Budući da je upravljačko-izvještajni informacijski sustav samo jedna od komponenti cijelog poslovnog informacijskog sustava, u ovom radu će ukratko biti objašnjena definicija informacijskog sustava, njegove komponente te životni ciklus. U radu su objašnjeni slojevi informacijskog sustava, s naglaskom na upravljački budući da je on u fokusu ovog rada. Nakon teorijskog dijela rada, dolazi prikaz i pojašnjenje implementirane simulacije upravljačko-izvještajnog informacijskog sustava na izoliranom dijelu poslovnog sustava fiktivnog proizvodnog poduzeća. Simulacija je implementirana uz pomoć Django razvojnog okvira u Pythonu, uz korištenje HTML, CSS te JavaScript programskih jezika u svrhu kreiranja osnovnog dizajna i grafičkih prikaza izvještaja.

Ključne riječi: informacijski sustav, upravljačko-izvještajni informacijski sustav, simulacija, upravljanje, izvještaj, Python, Django

Sadržaj

1. Uvod	1
2. Informacijski sustav.....	2
2.1. Komponente informacijskog sustava.....	2
2.1.1. Materijalno-tehničke komponente	2
2.1.2. Nematerijalne komponente	2
2.1.3. Ljudske komponente	3
2.1.4. Mrežne komponente.....	3
2.1.5. Organizacijske komponente	3
2.2. Životni ciklus informacijskog sustava.....	3
2.2.1. Faza inicijalizacije.....	4
2.2.2. Faza ekspanzije	4
2.2.3. Faza konsolidacije.....	4
2.2.4. Faza zrelosti	4
2.3. Uloga i utjecaj informacijskih sustava.....	5
3. Poslovni IS i upravljanje podacima	6
3.1. Funkcija poslovnog sustava	6
3.1.1. Karakteristike poslovnog sustava	6
3.2. Podatci i informacije	7
3.3. Pohrana podataka u obliku baze podataka.....	8
3.3.1. Modeliranje podataka	8
4. Slojevi poslovnog IS-a i njihova svrha	9
4.1. Izvršni sloj informacijskog sustava.....	10
4.1.1. Transakcijska obrada	10
4.1.2. Sustav za obradu transakcije	10
4.2. Upravljački sloj informacijskog sustava	11
4.2.1. Operativno upravljanje.....	12
4.2.2. Taktičko upravljanje.....	12
4.2.3. Strateško upravljanje.....	12
4.3. Suradnički sloj informacijskog sustava	13
5. Simulacija upravljačko-izvještajnog IS-a	14
5.1. Opis i ideja simulacije.....	14
5.2. Priprema simulacije	15
5.2.1. Alati i tehnologije.....	16
5.2.2. Pohrana podataka	17
5.3. Izrada i opis značajki simulacije	20
5.3.1. Kreiranje projekta	20
5.3.2. Kreiranje modela	20
5.3.3. Django Admin sučelje	21

5.3.4.	Početna stranica	22
5.3.5.	Unos u bazu podataka	22
5.3.6.	Pregled poduzeća	23
5.3.7.	Izvještaji	24
5.4.	Procesi na upravljačkoj razini simulacije.....	29
6.	<i>Zaključak</i>.....	31
	<i>Popis literature</i>	32
	<i>Popis slika</i>	33

1. Uvod

U ovom radu ću reći nešto o poslovnim informacijskim sustavima uz naglasak na upravljački informacijski sustav te njegovu izvještajnu komponentu. Objasnit ću važnost informacijskih sustava za poslovni sustav te uz izradu simulacije prikazati kako to otprilike izgleda u praksi.

Ovu temu sam odabrao kako bi zaključio trogodišnji preddiplomski studij Poslovni sustavi budući da sam na njemu naučio mnogo toga o poslovnim sustavima te kako izabrati i implementirati informacijske sustave u svrhu olakšanja poslovanja poduzeća. Glavni fokus rada bit će na upravljačkom informacijskom sustavu s izvještajnom komponentom jer vjerujem da, iako nije najviše razine, je vrlo važan za poslovanje poduzeća.

Za izradu simulacije odabrao sam programski jezik Python jer je on trenutno jedan od najpoželjnijih programskih jezika, a htio bih unaprijediti svoje znanje. Također ću koristiti web razvojni okvir Django koji je pisan u Python programskom jeziku jer znanje Django razvojnog okvira je korisno budući da se on koristi za izradu dinamičkih web aplikacija, a ima dobru mogućnost skalabilnosti.

2. Informacijski sustav

Prema Vargi (1994) informacijski sustav jest sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije vezane uz organizaciju i društvo, tako da budu dostupne i upotrebljive svakome kome su potrebne. Informacijski sustav aktivni je društveni sustav koji se može, ali ne mora, koristiti suvremenom informacijskom tehnologijom.

Kao što je u ranije navedeno, informacijski sustav prikuplja informacije koje je dužan obraditi kako bi ih pretvorio u čitljive informacije koje poslovni sustav može koristiti. Poslovni sustav može informacijskom sustavu predati povratnu informaciju ukoliko je došlo do neke pogreške ili odstupanja od stvarne situacije, a informacijski sustav tu povratnu informaciju može uvažiti u bilo kojem postupku u kojem je nastala pogreška ili odstupanje. Informacijski sustav u procesu obrade također pohranjuje informacije koje se koriste za prognoziranje budućih poslovnih stanja. (Stair, Reynolds, 2010.)

2.1. Komponente informacijskog sustava

U današnje vrijeme informacijskim sustavom se smatra sustav koji se sastoji od sljedećih komponenti koje su u stalnoj interakciji:

1. Materijalno-tehničke komponente
2. Nematerijalne komponente
3. Ljudske komponente
4. Mrežne komponente
5. Organizacijske komponente

2.1.1. Materijalno-tehničke komponente

Materijalno-tehničke komponente su poznate pod engl. nazivom *hardware* te njih čine strojevi, uređaji i sredstva koja su namijenjena isključivo ili pretežito za obradu podataka, odnosno informacija. Iz samog naziva doznajemo da se radi o fizičkim, ali neživim elementima informacijskog sustava. (Panian et al., 2010.)

2.1.2. Nematerijalne komponente

Nematerijalnu komponentu, engl. *software*, informacijskog sustava predstavlja ukupnost ljudskog znanja ugrađenog u strojeve, opremi i uređaje koje je samo po sebi predmet obrade ili diktira načinom obrade unutar sustava. Predmet obrade su su podatci od poslovne važnosti kao manifestacija ljudskog znanja raspoloživog u poslovnom informacijskom sustavu.

Metodologija takvog znanja je ugrađena u sustav u obliku računalnih aplikacija. (Panian et al., 2010.)

2.1.3. Ljudske komponente

Ljudsku komponentu, engl. *lifeware*, informacijskog sustava čine svi ljudi, tj. zaposlenici koji su unutar sustava imaju bilo kakvu funkciju ili na bilo koji način sudjeluju u radu sustava te se koriste rezultatom njegova rada. To mogu biti profesionalni informatičari koji djeluju u sustavu te je njihov udio puno manji u usporedbi s udjelom drugih ljudi unutar sustava, korisnika rezultata rada, kojih je uvijek više. (Panian et al., 2010.)

2.1.4. Mrežne komponente

Mrežna tj. prijenosna komponenta, engl. *netware*, informacijskog sustava sastoji se od komunikacijske infrastrukture za prijenos podataka između hardverskih jedinica unutar sustava ili s njegovim vezama unutar okoline. Pasivne elemente ove komponente čine razni oblici materijalnih i nematerijalnih komunikacijskih kanala i oni ne oblikuju podatke ni na koji način dok aktivni elementi, koje čine različiti namjenski, specijalizirani mrežni i komunikacijski uređaji, preoblikuju podatke prije, tijekom i nakon njihova prijenosa kako bi sam prijenos ili korištenje bilo učinkovitije. (Panian et al., 2010.)

2.1.5. Organizacijske komponente

Organizacijska komponenta, engl. *orgware*, informacijskog sustava predstavlja ukupnost standarda, mjera, postupaka i propisa kojima se funkcionalno i vremenski usklađuje rad prethodno navedenih komponenti sustava kako bi tvorile skladnu cjelinu, a to se naziva koordinacija, dok se vremensko usklađivanje naziva sinkronizacija. (Panian et al., 2010.)

2.2. Životni ciklus informacijskog sustava

Životni ciklus je opći koncept za upravljanje poslovnim informacijskim sustavom, a taj koncept nam je važan kako bi shvatili kako i u kojoj točki u vremenu dolazi do nastanka nekog informacijskog sustava, njegovog razvoja pa sve do njegove zamjene drugim informacijskim sustavom. (Panian et al., 2010.) Prema općem prikazu, postoje četiri faze životnog ciklusa informacijskog sustava, a to su:

1. Faza inicijalizacije (nastajanja)
2. Faza ekspanzije (rasta)
3. Faza konsolidacije (sazrijevanja)
4. Faza zrelosti sustava

2.2.1. Faza inicijalizacije

Faza inicijalizacije iliti nastajanja nekog informacijskog sustava započinje davanjem poticaja za razvitak novog poslovnog informacijskog sustava od strane menadžera i djelatnika poslovnog sustava koji se smatraju nedovoljno informiranim i vjeruju da im produktivnost opada. U ovoj fazi vode se različiti sastanci i razgovori kako bi se došlo do grube ideje budućeg sustava te sustav u ovoj fazi već počinje „živjeti“ iako još ne postoji, ali zato se stvaraju njegovi budući obrisi. (Panian et al., 2010.)

2.2.2. Faza ekspanzije

Nakon određenog vremena i ulaganja određenih sredstava, materijalnih i nematerijalnih, poslovni informacijski sustav dolazi u fazu ekspanzije u kojoj je uočljiv njegov kvantitativan rast u svim segmentima. U ovoj fazi se nabavlja oprema, pripremaju se programi te obrazuju djelatnici za njihovo korištenje, razvijaju se organizacijske metode i pribavljaju, tj. razvijaju mrežne komponente sustava. U ovoj fazi životnog ciklusa, krivulja je eksponencijalne prirode u smislu da je znatno rastuća. (Panian et al., 2010.)

2.2.3. Faza konsolidacije

Nakon što je sustav dosegnuo određenu razvojnu razinu koja na krivulji predstavlja točku infleksije, njegov razvoj se usporava kako bi se ostvarila potrebna kontrola nad njegovim radom i uspostavile željene mogućnosti upravljanja. Ova faza se naziva faza konsolidacije ili sazrijevanja zato što je sustav dovoljno zreo za korištenje, a dolazi do standardizacije određenih procesa i postupaka unutar sustava kako bi se postigao usklađen rad za budućnost tog sustava. Ovaj proces možemo nazvati „fino podešavanje“ iliti engl. *Fine Tuning*. (Panian et al., 2010.)

2.2.4. Faza zrelosti

Nakon što je sustav zadovoljio postavljenje kriterije kvalitete, učinkovitosti i djelotvornosti, on ulazi u fazu zrelosti kada sustav daje optimalne rezultate rada i stvara očekivanu poslovnu vrijednost. U cilju je da ova faza životnog ciklusa informacijskog sustava traje što duže kako bi uložena sredstva za razvoj informacijskog sustava bila dobro iskorištena. (Panian et al., 2010.)

2.3. Uloga i utjecaj informacijskih sustava

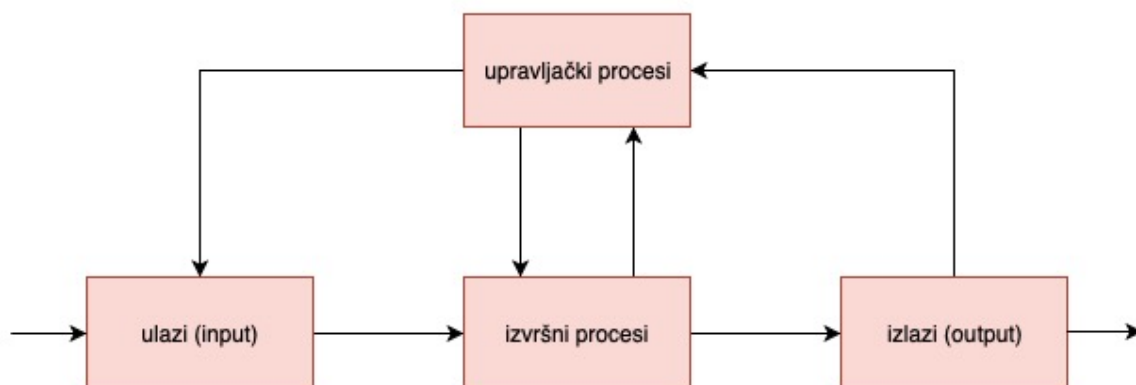
Razvitkom informacijskog društva, uloga informacijskih sustava postaje sve veća budući da se radi na informatizaciji gotovo svih aspekata poslovanja i života jer informacijski sustavi olakšavaju poslovanje i potiču rast konkurentnosti i efikasnosti. Povezanost između razvoja informacijskih sustavi i rasta produktivnosti je vidljiva u svim industrijskim granama. Elektronička trgovina je rezultirala intenzivnim povezivanjem ljudi i poduzeća putem interneta što je dovelo do novih načina povezivanja, novih poslovnih modela i novih, na webu temeljenih usluga, podržanih razvojem pametnih telefona i bežičnih tehnologija za prijenos podataka. Razvoj informacijskih sustava također je rezultirao padajućim cijenama softvera i hardvera te bržim i jeftinijim telekomunikacijskim uslugama što je dovelo do pojave brojnih start-up poduzeća koja predstavljaju porast poduzetničkih aktivnosti na globalnom tržištu. Uz taj razvoj također vidimo i porast usluga namijenjenih digitalnoj zabavi, poput streaming usluga za filmove, glazbu i video igre, a oni su zaslužni za novu kulturu društva u kojem trenutno živimo. (Pejić Bach et al., 2016.)

Intenzivnom upotrebom informacijskih sustava i informacijske tehnologije, poduzeću možemo donijeti brojne strateške koristi koje utječu na konkurentsku poziciju, ali samo ako ciljevi njezine upotrebe podržavaju i proširuju strateške ciljeve poslovanja. Poslovni informacijski sustavi izravno utječu na konkurentnost poslovanja na dva načina: pozitivno utječu na operativnu efikasnost poslovanja (podupiru strategiju niskih troškova) ili u određenim okolnostima postaju pokretači inovativnosti i promjena u poslovanju (podupiru strategiju razlikovanja ili diferencijacije poslovanja). (Pejić Bach et al., 2016.)

3. Poslovni IS i upravljanje podacima

Kao što je ranije navedeno, informacijski sustav je skup komponenti koje zajedno čine sustav koji radi u svrhu ostvarivanja nekog cilja.

3.1. Funkcija poslovnog sustava



Slika 1. Funkcije poslovnog sustava

Na slici iznad možemo vidjeti funkcije poslovnog sustava. Svrha poslovnog sustava je proizvesti izlaze ili outpute tako da ulaze ili inpute obrađuje izvršnim procesima. Postoji povratna veza od izlaza koja se temelji na povratnoj informaciji ili feedback-u. Ta povratna informacija je ključna upravljačkim procesima kako bi mogli kontrolirati izvođenje izvršnih procesa za ostvarenje regulacije poslovnog sustava. (Panian et al., 2010.) U slučaju proizvodnje nekog proizvoda, ukoliko se na tržištu osjeti veća potražnja, ta informacija odlazi upravljačkom mehanizmu kako bi tu informaciju odrazio na ulazima te na drugim sustavima koji djeluju u okolini nekog poslovnog sustava.

3.1.1. Karakteristike poslovnog sustava

Komponente sustava zajednički rade u ostvarenju kolektivnog cilja te je cilj cijelog poduzeća ostvariti dobit. Sustav nikad nije izoliran od okoline te u okolini djeluju i drugi sustavi. Poduzeća moraju dobro odrediti granice sustava kako ne bi došlo do zabune. Veći sustavi su složeni od više jednostavnijih sustava što znači da svaki sustav možemo raščlaniti na podsustave što znači da postoji hijerarhijska organiziranost. (Panian et al., 2010)

3.2. Podatci i informacije

Podatkom zapisujemo određenu činjenicu te podatak možemo zapisati na različite medije na način koji je primjeren za određeni medij. Zapisane podatke moramo pročitati i interpretirati kako bi iz njih dobili informaciju, odnosno obavijest. Dobivene informacije možemo zapisivati u obliku podataka tako da vrijedi obrnuto. (Panian et al., 2010)

Elementarna informacija se dobiva interpretacijom elementarnog podatka, a on je znakovni prikaz jednog obilježja nekog objekta. Primjerice, ako se podatak „Ivan“ odnosi na obilježje imena neke osobe, tada ga interpretiramo kao ime osobe, ali ako se radi o nazivu poduzeća ili slično, onda ga interpretiramo drugačije. Informacije nam donose novosti što znači da nas obavještavaju o zbivanjima oko nas te o zbivanjima koja su važna za poduzeće. Ljudske odluke ne mogu biti bolje od informacija na kojima se temelje tako da za kvalitetne odluke, potrebne kvalitetne informacije. Svaka kvalitetna informacija je točna, potpuna, relevantna te pravovremena. Točna informacija ispravno opisuje stanje stvari. Potpuna informacija u cijelosti i objektivno opisuje stanje stvari. Relevantna informacija je ona informacija koja odgovara problemu koji zahtjeva odlučivanje i osobi koja odlučuje, a svaka pravovremena informacija je ona informacija koja je dobivena na vrijeme. (Panian et al., 2010.)

Vrijednost informacije je teško utvrditi budući da se njezina važnost pokazuje u svakom trenutku u kojemu je potrebna. Vrijednost informacije je najveća ako se nalazi u području prognoziranja budući da su nam tad pravovremene informacije važne prije nego što dođe do procesa donošenja odluke. Vrijednost je nešto manja u području stvarnog vremena i najmanja u području zastarijevanja budući da je tad najčešće već prekasno za neku informaciju. (Panian et al., 2010.)

Znanje je kombinacija podataka i informacija kojemu je dodano ekspertno mišljenje, vještina i iskustvo. Znanjem određujemo kako koristiti određene informacije. Znanje je važno za proces donošenja odluka budući da je svaka dobra odluka temeljena na znanju. Korištenje i razvoj poslovnih informacijskih sustava zahtjeva znanje o samom poslovnom sustavu te o njegovim komponentama budući da bez znanja nije moguće raditi ili upravljati niti jednom djelatnosti. (Panian et al., 2010)

3.3. Pohrana podataka u obliku baze podataka

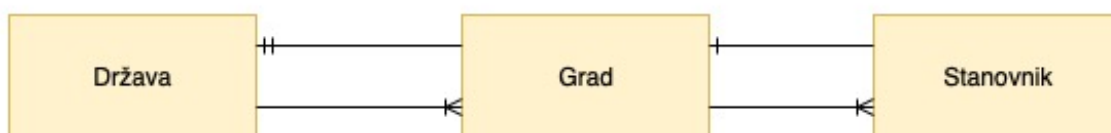
U današnje vrijeme poduzeća se sastoje od više elemenata koji rukuju s previše podataka koje je važno sigurno pohraniti. Najčešći oblik pohrane su baze podataka koje nude jednostavno i sigurno pohranjivanje podataka. Važno je odrediti koji podatci su relevantni za poslovanje te baza mora imati mogućnost jednostavnog korištenja i modificiranja podataka koji su već u bazi. (Stair, Reynolds, 2010.) Prilikom izgradnje baze podatka važno je pripaziti na ova pitanja:

1. Sadržaj – koji podatci će se prikupljati i po kojoj cijeni?
2. Pristup – koji podatci će biti dostupni kojim korisnicima i kada?
3. Logička struktura – kako će podatci biti pohranjeni da to ima logičkog smisla korisniku?
4. Fizička organizacija – gdje će podatci biti fizički pohranjeni?

Kao što je ranije navedeno, informacijski sustav je skup komponenti koje zajedno čine sustav koji radi u svrhu ostvarivanja nekog cilja.

3.3.1. Modeliranje podataka

Logička struktura iliti logički dizajn podataka u bazi podataka je vrlo važan i on mora odgovarati načinu na koji korisnici koriste podatke. Logički dizajn je apstraktan model podataka u kojemu se identificiraju veze između podataka koji se potom grupiraju u grupe koje imaju smisla s obzirom na primjenu podataka. Važno je da ne dolazi do redundantnosti podataka budući da to može naškoditi performansama informacijskog sustava koji se koristi tom bazom podataka. Za ostvarenje logičkog dizajna najčešće koristimo ER dijagram iliti engl. entity-relationship diagram koji na jednostavan način, uz pomoć osnovnih grafičkih simbola, prikazuje veze između podataka. Na grafičkom prikazu, kvadrati najčešće prikazuju podatke ili tablice s podacima dok crtice između njih pokazuju vrstu veze s kojom su podatci povezani. (Stair, Reynolds, 2010.)

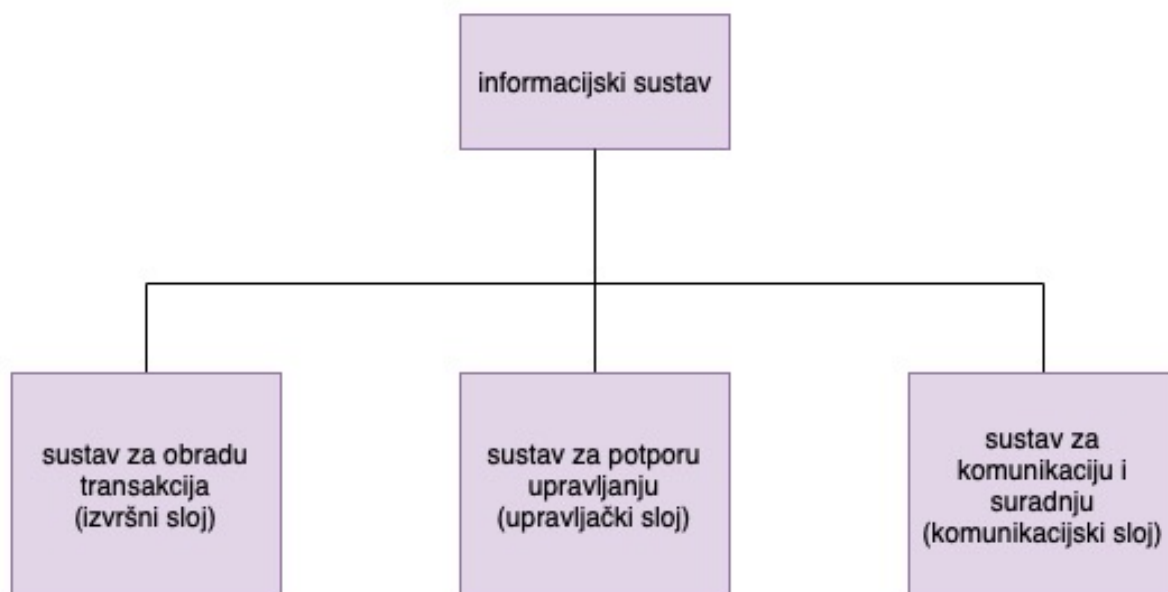


Slika 2. Primjer ER dijagrama

4. Slojevi poslovnog IS-a i njihova svrha

Možemo zaključiti da je poslovanje unutar poslovnog sustava temeljeno na podacima te da postoje različiti poslovni procesi, npr. izvršni i upravljački. Svaki od tih procesa možemo svrstati pod istoimeni podsustav. (Panian et al., 2010.) Poslovni informacijski sustav poduzeću može olakšati poslovanje na tri načina:

1. Izvršnom dijelu sustava može pomoći s izvršenjem poslovnih procesa, a taj sloj informacijskog sustava nazivamo sustavom za obradu transakcija
2. Upravljačkom dijelu sustava može olakšati upravljanje poslovnim sustavom, a taj sloj informacijskog sustava nazivamo sustavom za potporu upravljanju
3. Poslovnom sustavu u cjelini može omogućiti bolju komunikaciju sa samim sobom i s okolinom sustava, a taj sloj informacijskog sustava nazivamo sustavom za komunikaciju i suradnju.



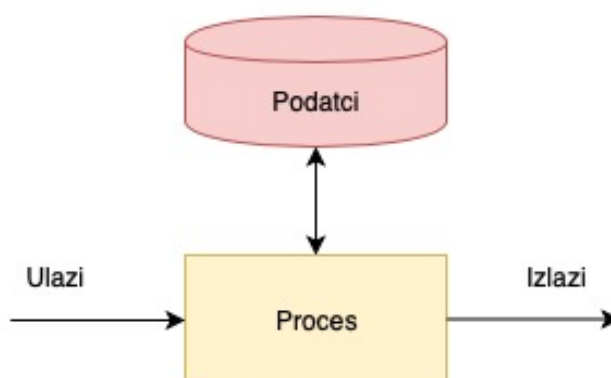
Slika 3. Slojevi poslovnog informacijskog sustava

4.1. Izvršni sloj informacijskog sustava

Izvršni sloj informacijskog sustava je zaslužan za izvođenje izvršnih poslovnih procesa, tj. onih procesa koji određene ulaze pretvaraju u željene izlaze. Uvođenjem informacijskog sustava, tj. procesom informatizacije, želimo postići što veću razinu automatizacije poslovnih procesa. Neke poslovne procese nije moguće automatizirati na način na koji su radili do tada te je potrebno izvršiti reinženjering poslovnog procesa kako bi ga poboljšali na način koji je izvediv za automatizaciju. (Panian et al., 2010)

4.1.1. Transakcijska obrada

Izvršni procesi imaju transakcijski karakter te ih zato možemo nazvati transakcijskim procesima. Transakcijski proces uzima materijalni ili informacijski ulaz koji pretvara u izlaz tj. output. Ukoliko je potrebno pamtit stanje tog procesa, to stanje se bilježi u podacima. Prema tome, izlaz tj. output čine prerađeni ulazi tj. inputi, a izlazi također mogu biti materijalne ili informacijske prirode. (Panian et al., 2010.)



Slika 4. Transakcijski proces

4.1.2. Sustav za obradu transakcije

Sustav za obradu transakcija je sustav koji pruža potporu tekućem izvođenju poslovnog procesa izvođenjem niza transakcija. Transakcijski sustav ima tri funkcije:

1. Izvršava transakcije i vodi evidenciju o obavljenim transakcijama
2. Stvara dokumente vezane uz transakcije nužne za poslovanje
3. Izvještava o stanju poslovanja (ova komponenta je važna za rad upravljačko-izvještajnog sustava)

4.2. Upravljački sloj informacijskog sustava

Upravljanje poslovnim sustavom uključuje planiranje, organiziranje te kontroliranje aktivnosti poslovnog sustava, a temelji se na donošenju odluka. Za donošenje dobrih poslovnih odluka važno je imati kvalitetne informacije što nam pokazuje važnost izgradnje dobrih poslovnih informacijskih sustava za postojeći poslovni sustav. (Panian et al. 2010.)

Odlučivanje je proces koji se obavlja kroz sljedeće korake:

1. Prepoznavanje problema – uključuje shvaćanje i definiranje problema, pronalaženje izvora informacija, prikupljanje i procjena vrijednosti informacija
2. Oblikovanje opcija rješenja – uključuje formiranje mogućih rješenja problema koji su u skladu s ciljem poslovnog sustava
3. Odabir najpovoljnije opcije – donositelj odluke odabire rješenje koje je najbolje za rješavanje određenog problema poslovnog sustava
4. Provedba odabrane opcije – uključuje provođenje određenih mjera i aktivnosti koje su u skladu s odabranom opcijom
5. Ocjena provedbe – uključuje ispitivanje i analizu provedbe te utvrđuje je li provedba riješila problem poslovnog sustava

Informacije koje se koriste pri odlučivanju mogu biti iz različitih izvora. Unutarnji podatci nastaju unutar samog poslovnog procesa, dok vanjski podatci nastaju izvan poslovnog sustava. Primjer unutarnjih podataka mogu biti povratne informacije o nekom od procesa ili informacije o stanju na skladištu dok primjer za vanjske podatke može biti stanje na tržištu ili općenito informacije o sociopolitičkoj klimi izvan poslovnog sustava. (Panian et al., 2010.)

Cilj upravljačkog sloja informacijskog sustava je isporučiti kvalitetne informacije za donošenje odluka svim onima kojima je to potrebno. To može uključivati radnike poput menadžera i raznih stručnjaka. Takvo odlučivanje se decentralizira i demokratizira jer se u proces poslovnog odlučivanja uključuje sve veći broj sudionika. Donositelji odluka često donose odluke u nepovoljnim uvjetima kao što su nedostatak vremena ili manjak stručnog znanja, a njihove spoznajne mogućnosti su također ograničene. Zbog toga im je potrebna identifikacija, pronalaženje i analiza relevantnih podataka odnosno informacija, a tu mogućnost je lakše ostvariti uvođenjem upravljačkog informacijskog sustava. (Panian et al., 2010.)

Postoje tri tipična oblika upravljanja koja će biti objašnjena u nastavku.

4.2.1. Operativno upravljanje

Operativno upravljanje je tip upravljanja kojim se bave operativni ili niži menadžeri koji vode i nadgledaju poslovne procese na operativnoj razini i provode odluke taktičkih menadžera. Njihove odluke se često ponavljaju, a odlučivanje je racionalno i strukturirano. Kod strukturiranog odlučivanja postupak odlučivanja je određen prema ranije poznatim pravilima, procedurama ili algoritmima koji su temeljeni na eksplicitnim podacima. Strukturirano odlučivanje je već sadržano u mnogim transakcijama koje su automatizirane budući da se neke transakcije neće izvršiti ukoliko nisu u okviru pravila i procedura koje su potrebne za izvršavanje. Sustav za obradu transakcija obično sadrži izvještajne transakcije koje nam omogućuju izvještavanje temeljeno na podacima iz transakcije. Njih implementiramo pomoću raznih programskih jezika, a pokreću se prema zahtjevu i potrebama korisnika, vremenskom planu ili na određeni okidač koji je ranije postavljen. (Pejić Bach et al., 2016.)

4.2.2. Taktičko upravljanje

Taktičko upravljanje je upravljanje kojim se bavi taktički ili srednji menadžeri koji nadgledaju i prate skupine poslovnih procesa za koje su zaduženi. Odlučivanje kod taktičkog upravljanja je uglavnom polu-strukturirano jer se ne može uvijek unaprijed utvrditi pravilom ili procedurom. Srednjem menadžmentu je pri odlučivanju potrebno stručno znanje, umješnost i iskustvo. Za analizu su im potrebne agregirane informacije u obliku sumarnih periodičkih izvještaja ili izvještaja o izuzecima u dnevnim aktivnostima. Takve izvještaje nije moguće unaprijed implementirati i programirati jer su nepredvidivi. Iterativno analiziraju podatke, a analitičar postavlja usmjerene upite bazi ili skladištu podataka te potom analizira rezultate dok ne dijagnosticira problem. (Pejić Bach et al., 2016)

4.2.3. Strateško upravljanje

Strateškim upravljanjem bave se menadžeri najviše razine, a oni su ti koji donose dugoročne strateške odluke za poslovanje poduzeća. Strateško upravljanje se oslanja na nestrukturirano odlučivanje jer kod njega nije moguće propisati pravilo ili proceduru odlučivanja. Teret odlučivanja je na osobi koja odlučuje, a zadatak informacijskog sustava je da toj osobi pruži relevantne informacije koje će joj pomoći pri donošenju odluke. Nije moguće predvidjeti potrebne informacije. Kod strateškog upravljanja dolazi do primjene dubinske analize podataka koja je temeljena na statistici, matematici i umjetnoj inteligenciji te ona daje najbolje informacije koje su potrebne menadžerima najviše razine. (Pejić Bach et al., 2016.)

4.3. Suradnički sloj informacijskog sustava

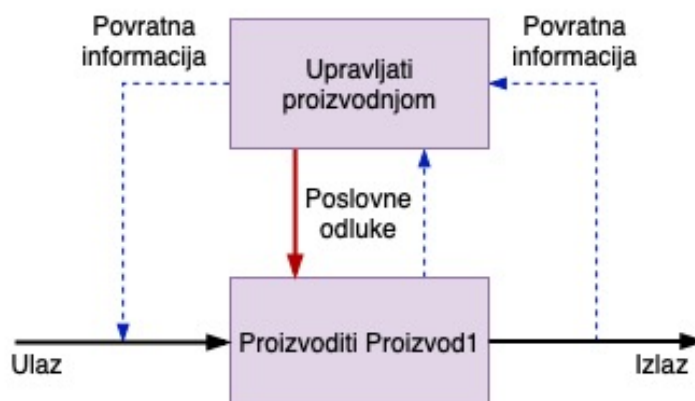
Suradnički sloj informacijskog sustava kao što mu ime govori, zaslužan je za suradnju i komunikaciju unutar poslovnog sustava. U poslovnom sustavu sudjeluje velik broj sudionika, a njihova koordinacija i predanost cilju je vrlo važna, a suradnički sloj informacijskog sustava olakšava njihovu međusobnu interakciju. Tu se često radi o aplikacijama za analizu podataka, prezentiranje, kreiranje izvještaja i drugih sličnih dokumenata. Razvojem tehnologije i chat sustava, dolazi do boljih tehnologija poput VOIP koje znatno olakšavaju komunikaciju unutar poslovnog sustava. Uredski alati poput Microsoft Office i OpenOffice su danas pristupačni čak i za manje informatički pismene korisnike što ih čini odličnima za poslovne sustave sa zaposlenicima koji nisu striktno informatičari. (Panian et al., 2010.)

5. Simulacija upravljačko-izvještajnog IS-a

Prema Hrvatskoj enciklopediji, riječ simulacija dolazi od latinskog pojma *simulatio* što znači pretvaranje ili prijevara. Simulacija ima više značenja, ali u znanosti i tehnici pa tako i u informatičkom smislu, ona je postupak kojim se ponašanje nekog objekta ili odvijanje neke pojave istražuje na fizičkom ili računalnome modelu. Obično se provodi u edukacijske svrhe, ali i onda kad je rad na stvarnom objektu previše zahtjevan, skup ili neetičan ili pak kada pravi sustav još ne postoji pa se ne može promatrati. (Hrvatska enciklopedija, 2019.)

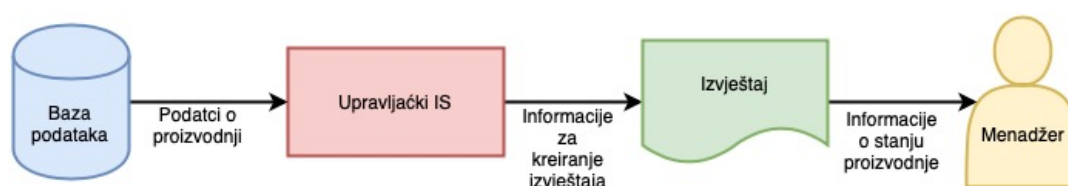
5.1. Opis i ideja simulacije

Simulacija je implementirana na temelju fiktivnih podataka o fiktivnom poduzeću koje je smišljeno u svrhu izrade ovog rada te svaka sličnost sa stvarnim sustavom, poduzećem ili osobom je slučajna. Budući da su upravljački sustavi obično velikog opsega te su implementirani nad cijelim poslovnim sustavima kako bi pomogli u izvršenju i poboljšanju poslovnih procesa unutar sustava, u svrhu ovog rada, simulacija je implementirana na izoliranom dijelu poslovnog sustava nad procesom „Proizvoditi proizvod“. Radi se o fiktivnom poduzeću koje se bavi proizvodnjom Proizvoda1 koji se proizvodi u dva proizvodna pogona, Pogon1 i Pogon2. Svaki pogon ima određeni broj radnika što naravno utječe na ukupnu produktivnost čitavog pogona. Glavni cilj simulacije je proizvesti informaciju uz pomoć koje je vidljiv rezultat proizvodnje na način koji je lako razumljiv i manje informatički obrazovanim osobama.



Slika 5. Prikaz promatranog procesa

Na slici 5. je prikazan proces koji će se promatrati u ovoj simulaciji. Crvenom strelicom je prikazan upravljački tok, plavom strelicom su prikazani informacijski tokovi, a crnom strelicom je prikazan materijalni tok ovog procesa. Upravljački informacijski sustav u srži funkcionira tako da na temelju podataka, koji su kreirani i pohranjeni od strane transakcijskog informacijskog sustava, stvara izvještaje koji su prikladni za korištenje u upravljačkom sloju poslovnog informacijskog sustava. Izvještaji u ovoj simulaciji će biti prikazani kao uređeni i formatirani ispis podataka iz baze podataka uz popratni grafički prikaz.



Slika 6. Prikaz upravljačkog sustava

Na slici 6. prikazan je princip rada upravljačko-izvještajnog informacijskog sustava čija je simulacija implementirana u ovom radu. Kao što je ranije navedeno, upravljački informacijski sustav obrađuje informacije iz baze podataka koje je kreirao transakcijski informacijski sustav te na temelju istih kreira izvještaje koji služe kao pomoć pri donošenju odluka u upravljačkom ili menadžerskom sloju poslovnog sustava.

5.2. Priprema simulacije

Važan korak u izradi simulacije je priprema iste budući da bez odgovarajućih alata i tehnologija nije moguće ideju implementirati u obliku programskog rješenja. Izradu simulacije upravljačko-izvještajnog sustava odlučio sam implementirati koristeći programski jezik Python jer je on jedan od trenutno najpoznatijih i najboljih rješenja kad je u pitanju razvoj softvera ili nekakvih brzih rješenja za potrebe obrazovanja. Python se koristi za izradu web stranica, umjetnu inteligenciju, strojno učenje te u razne obrazovne i znanstvene svrhe budući da posjeduje brojne module koji olakšavaju isto. (Python Software Foundation, 2019.)

5.2.1. Alati i tehnologije

Prilikom implementacije simulacije upravljačko-izvještajnog informacijskog sustava korišten je Django razvojni okvir koji se koristi za izradu web aplikacija. Django je potpuno besplatan i open-source te je trenutno jedan od najpoznatijih razvojnih okvira za razvoj dinamičkih web stranica. Django je poznat po tome da dolazi uz mnoge module tako da pri razvoju nije potrebno pisati redundantni kod ili programirati neke osnovne koncepte iz temelja budući da su već napisani unutar samog Djanga. Django je znatno brži za razvoj od sličnih razvojnih okvira, upravo zbog svih mogućnosti koje dolaze uz njega. Neke od najvažnijih stvari koje dolaze uz Django su ovjera korisnika, administracija sadržaja, mapiranje stranica te mnoge druge mogućnosti koje je inače potrebno ručno razvijati. Jedna od najvećih prednosti Djanga naspram sličnih razvojnih okvira je njegova skalabilnost što znači da može podnijeti više prometa bez gubitka na performansama. Već se da zaključiti da je Django vrlo moćan razvojni okvir te ga koriste mnoge kompanije, vlade i organizacije za izgradnju raznih sustava te iako je u ovom slučaju korišten za izradu simulacije, itekako je moguće s njime implementirati pravu verziju sustava. (Django Software Foundation, 2019.)

Za izradu grafičkih prikaza u obliku grafova i raznih grafikona korišten je JavaScript modul pod nazivom Chart.js koji je trenutno jedan od najpoznatijih modula za istu namjenu. Dolazi u dvije inačice, jedna od njih je klasična verzija koja je distribuirana koristeći Node Package Manager poznat pod kraticom npm te CDN (engl. content delivery network) inačica koja je korištena za implementaciju simulacije budući da ju nije potrebno preuzimati već se može uključiti unutar HTML datoteke kao i sve ostale JavaScript datoteke ili alati.

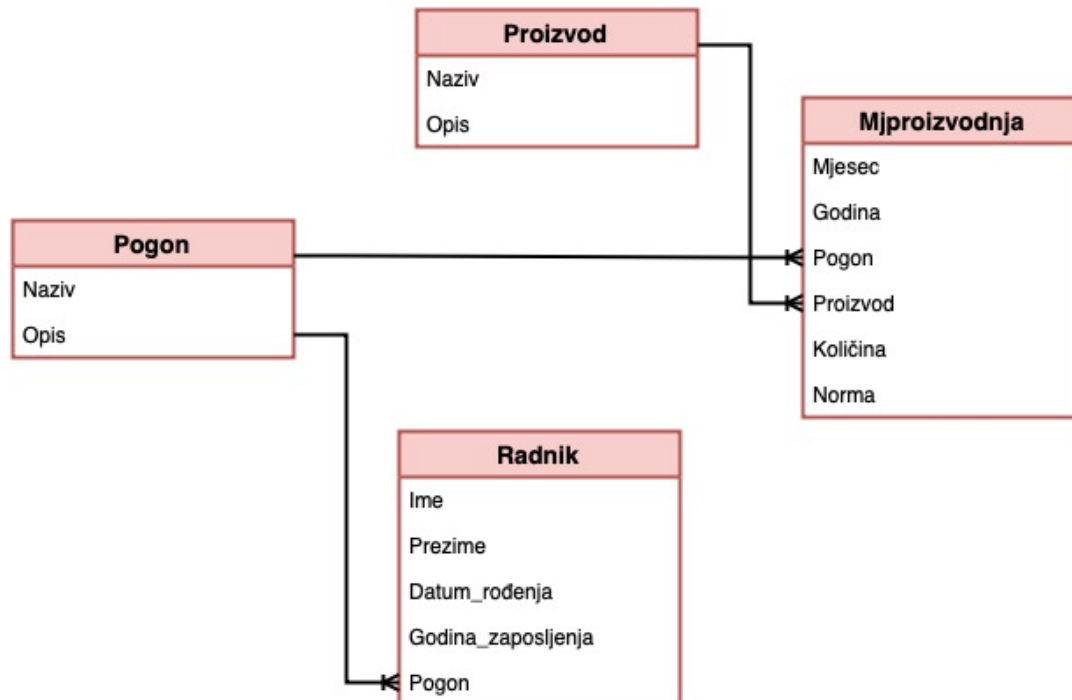
Za sam razvoj korišten je Visual Studio Code koji je trenutno najbolji alat za pisanje koda budući da dolazi uz mnoge ekstenzije koje olakšavaju razvoj aplikacija i web stranica. Važno je napomenuti da je od velike pomoći bila Django ekstenzija budući da dolazi uz Django-HTML tip dokumenta koji koristi specifične naredbe koje se razlikuju od standardnog HTML-a.

Budući da se naredbe za Django razvojni okvir pokreću kao klasične python datoteke, važno je napomenuti korištenje zsh Unix ljuske unutar Terminal.app aplikacije. Također je korišten git kao alat za upravljanje izvornim kodom jer prilikom pisanja koda uvijek može doći do neke veće greške, a najlakše je povući zadnju stabilnu verziju koda s nekog od servisa za git, u ovom slučaju GitHub-a.

5.2.2. Pohrana podataka

Za pohranu podataka odlučio sam koristiti klasičnu bazu podataka budući da s njima imam iskustva te vjerujem da SQL baze podataka imaju puno više prednosti od nekih noSQL baza podataka ili pak datoteka strukturiranog sadržaja. Pri implementaciji projekta korišten je SQLite sustav za relacijske baze podataka zato što on dolazi u paketu s Django razvojnim okvirom te je on najkorišteniji sustav za upravljanje relacijskim bazama podataka. SQLite je napisan u C programskom jeziku što jamči njegovu brzinu i jednostavnost korištenja, a kompatibilan je sa svim razvojnim okvirima i programskim jezicima. Jedna od najvećih prednosti SQLite-a naspram drugih sustava za upravljanje relacijskim bazama podataka je to što radi bez servera na način da podatke zapisuje unutar jedne datoteke što je praktično za izradu manjih aplikacija kojima nije potrebna snaga klasičnog sustava za upravljanje relacijskim bazama podataka poput PostgreSQL ili MySQL.

Kao što je ranije navedeno, u implementaciji simulacije se radi na izoliranom procesu na temelju kojeg su kreirani podatci potrebni za njegovo izvršavanje te simulacija sadrži četiri tablice pod nazivima: Radnik, Pogon, Proizvod i Mjproizvodnja (puno značenje: mjesečna proizvodnja).



Slika 7. ER model simulacije

Django koristi moćni Python ORM (engl. Object-relational mapping) što znači da nije potrebno pisati SQL već samu implementaciju baze možemo napisati koristeći Python programski jezik budući da za svaku SQL naredbu, postoji njemu ekvivalentan izraz u programskom jeziku. Django je osnovan na MVC (model - view - controller) arhitekturi što znači da se „model“ dio odnosi na bazu podataka, dok „controller“ dio manipulira podacima unutar „modela“ te stvara prikaz iliti „view“ u obliku ranije napisanih predložaka gdje filtrira i prikazuje podatke te radnje s njima, onako kako mi to želimo.

Modeli, iliti tablice, se kreiraju unutar datoteke `models.py` koja se nalazi u direktoriju Django projekta. Nakon kreiranja modela, potrebno je provesti migracije kako bi se kreirani modeli primijenili unutar aplikacije.

Ovako to izgleda u `models.py` datoteci:

```
class Pogon(models.Model):
    naziv = models.CharField(max_length=20)
    opis = models.TextField()

    class Meta:
        verbose_name_plural="Pogoni"

    def __str__(self):
        return self.naziv

class Proizvod(models.Model):
    naziv = models.CharField(max_length=20)
    opis = models.TextField()

    class Meta:
        verbose_name_plural="Proizvodi"

    def __str__(self):
        return self.naziv

class Radnik(models.Model):
    ime = models.CharField(max_length=20)
    prezime = models.CharField(max_length=30)
    datumrođenja = models.DateField()
    godzaposljenja = models.IntegerField()
    pogon = models.ForeignKey(Pogon, on_delete=models.CASCADE)

    class Meta:
        verbose_name_plural="Radnici"

    def __str__(self):
        return '{} {}'.format(self.prezime, self.ime)
```

```

class Mjproizvodnja(models.Model):
    mjesec = models.CharField(max_length=20)
    godina = models.IntegerField()
    pogon = models.ForeignKey(Pogon, on_delete=models.CASCADE)
    proizvod = models.ForeignKey(Proizvod, on_delete=models.CASCADE)
    kolicina = models.IntegerField()
    norma = models.IntegerField()

    class Meta:
        verbose_name_plural="Mjesečna proizvodnja"

    def __str__(self):
        return '{} / {} / {}-{}'.format(self.pogon, self.proizvod,
self.mjesec, self.godina)

    def razlika(self):
        return (self.norma - self.kolicina)

```

ORM funkcionira tako da za svaku tablicu kreiramo Python klasu na temelju Model predloška koji je uvezen iz `django.db` modula. Unutar klase zadajemo attribute, slično kao što to radimo koristeći SQL, ali u slučaju ORM-a koristimo drugačije nazive atributa.

Unutar klase možemo kreirati podklase i funkcije uz pomoć kojih manipuliramo sadržajem klase kako bi dobili željene rezultate. Klasa „Meta“ se kreira uz opciju `verbose_name_plural` te nju koristimo kako bi tablicu prikazali sa željenim nazivom umjesto onog naziva koji smo zadali pri kreiranju nadklase. Ta opcija je vrlo praktična kad se radi o tablicama s više riječi u nazivu. Funkcija `__str__` nam omogućava promjenu načina prikaza unosa u bazu podataka unutar sustava što je vrlo važno ako uz pomoć administracijskog sučelja pregledavamo sve unose u bazu podataka. Dodatne funkcije poput funkcije `razlika` u tablici `Mjproizvodnja` služe za manipulaciju podacima koji se nalaze u pojedinom unosu. U ovom slučaju funkcija `razlika` vraća razliku između norme i količine kako bi dobili izračunatu vrijednost manjka proizvodnje koja nam je potrebna kasnije u kreiranom izvještaju.

5.3. Izrada i opis značajki simulacije

U nastavku ću ukratko opisati svaku od implementiranih značajki simulacije na način da objasnim kako radi i ukratko prikažem način na koji je ta značajka implementirana koristeći Django i popratne alate.

5.3.1. Kreiranje projekta

Koristeći komandno sučelje (engl. command line interface, u nastavku CLI) moramo ući u željeni direktorij i pokrenuti naredbu `django-admin startproject mis_sim` kako bi kreirali projekt `mis_sim` te početne datoteke potrebne za daljnji rad na istom. Nakon kreiranja projekta, potrebno je kreirati zasebnu aplikaciju budući da u Django razvojnem okviru svaka web aplikacija je zasebna aplikacija unutar projektnog direktorija, a to je potrebno kako bi preglednost projekta bila bolja te kako bi mogli koristiti značajke iz jedne aplikacije u drugima iako nisu direktno povezane. Zasebnu aplikaciju pod nazivom `mis` kreiramo s naredbom `python manage.py startapp mis`. Kreiranu aplikaciju potrebno je uključiti u projekt kako bi ju Django prepoznao, a to radimo tako da dodamo našu aplikaciju unutar `settings.py` datoteke u glavnom direktoriju projekta:

```
INSTALLED_APPS = [  
    . . .  
    'mis.apps.MisConfig',  
    . . .  
]
```

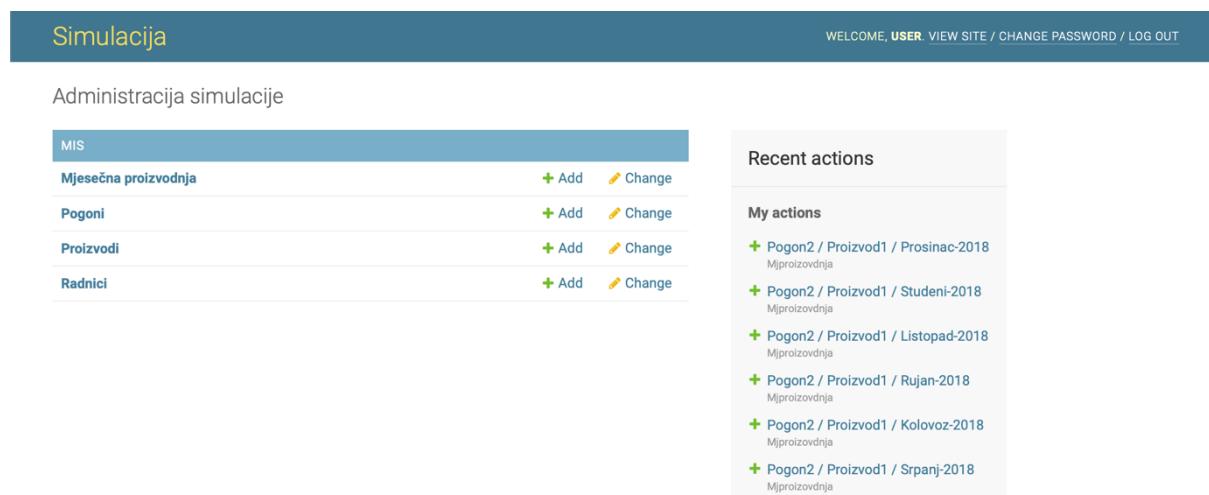
Nakon što je kreirano sve što je potrebno za projekt, potrebno je pokrenuti komandu `python manage.py runserver` kako bi pokrenuli lokalni server na kojem možemo vidjeti rezultat našeg rada, točnije. implementacije određenih značajki.

5.3.2. Kreiranje modela

Kreiranje modela je najvažniji dio Django projekta budući da se na njima temelji čitava aplikacija. Modeli potrebni za ovu simulaciju su objašnjeni ranije u poglavlju vezanom uz pohranu podataka, ali to nije sve budući da nakon kreiranja modela potrebno je pokrenuti naredbu `python manage.py makemigrations` kako bi kreirali migracije koje su vezane uz naše modele. Nakon što smo kreirali migracije, potrebno ih je primijeniti uz pomoć naredbe `python manage.py migrate`. Važno ih je uključiti u Django Admin sučelje uz pomoć naredbe `admin.site.register`.

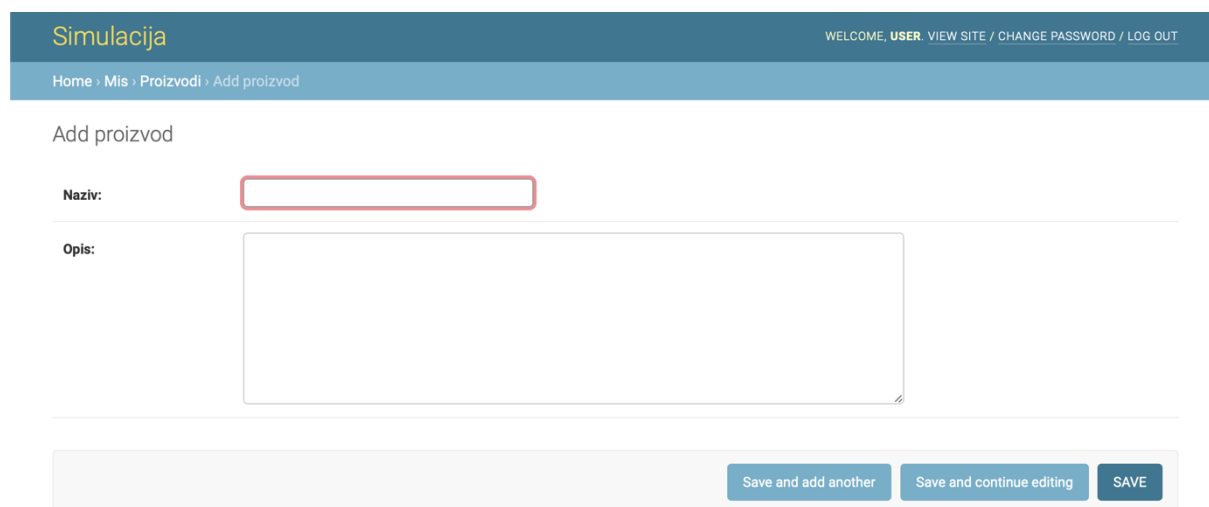
5.3.3. Django Admin sučelje

Django Admin je moćno administracijsko sučelje koje dolazi uz Django, a omogućava CRUD mogućnosti za sve prethodno registrirane modele. Budući da je fokus ove simulacije na upravljačko-izvještajnim komponentama, redundantno je dodatno kreirati elemente transakcijskog informacijskog sustava jer ih možemo postići uz korištenje Django Admin sučelja. Dolazi uz verifikaciju i inače se koristi za unutarnji menadžment, tj. za one kojima je odobreno da uređuju i unose podatke u bazu podataka.



Slika 8. Administracija simulacije

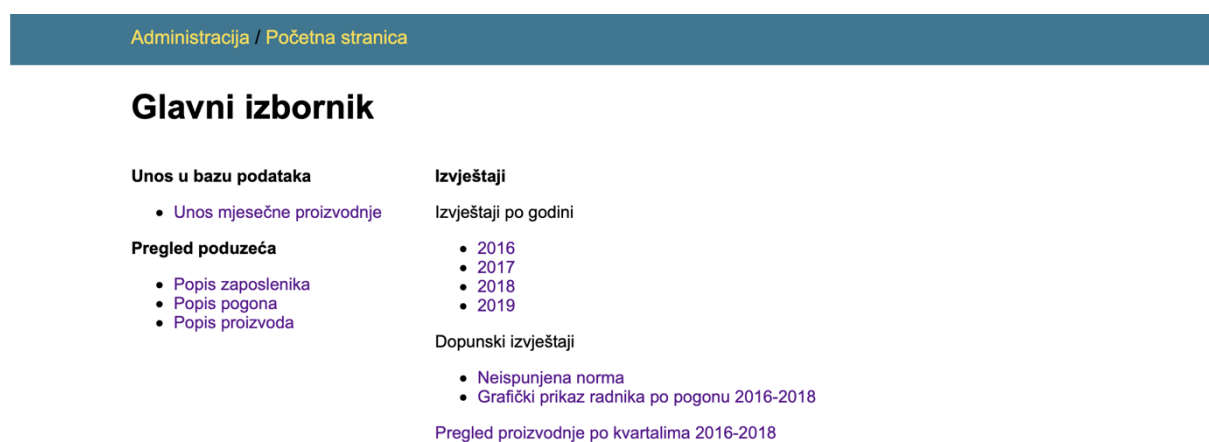
Na slici 8. je prikazana administracija simulacije kreirana uz pomoć Django Admin sučelja. Možemo vidjeti nedavne promijene te možemo pristupiti nekoj od tablica unutar baze podataka kako bi unijeli nove podatke i napravili nove transakcije. Na slici 9. možemo vidjeti primjer forme za unos novog proizvoda u bazu podataka.



Slika 9. Forma za dodavanje proizvoda

5.3.4. Početna stranica

Na slici 10. je prikazana početna stranica simulacije gdje je vidljiv glavni izbornik s mogućnostima koje su implementirane unutar simulacije. Možemo joj pristupiti koristeći „view site“ mogućnost unutar administracijskog sučelja, ali ona se također i otvara automatski nakon pokretanja aplikacije. U zaglavlju se nalazi izbornik s Administracijom i Početnom stranicom koja korisnika vraća uvijek na početnu stranicu s glavnim izbornikom. Svaka od ponuđenih značajki unutar glavnog izbornika je hiperveza na istoimenu značajku. Kroz sljedećih nekoliko poglavlja će biti objašnjene značajke prikazane na slici 10.



Slika 10. Prikaz početne stranice

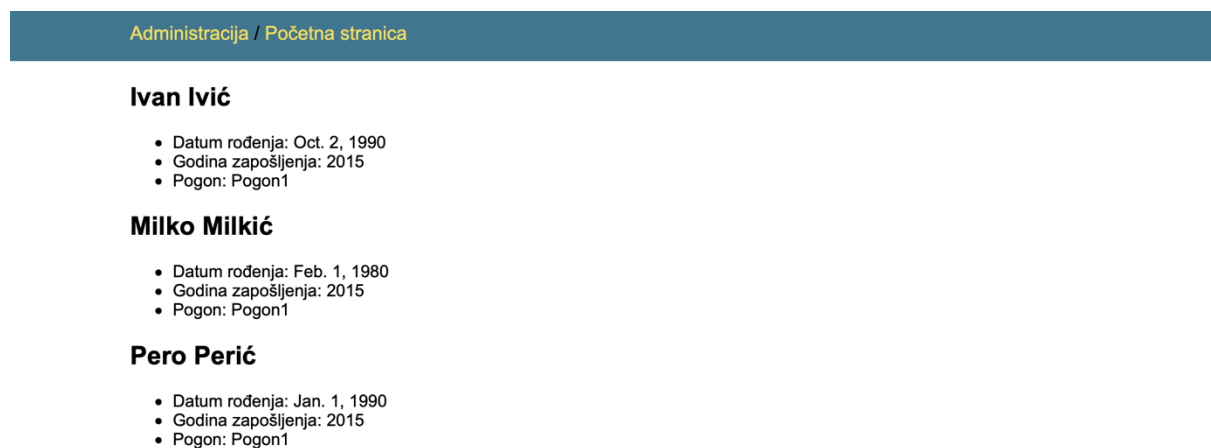
5.3.5. Unos u bazu podataka

Na slici 11. nalazi se forma za unos mjesečne proizvodnje u bazu podataka te iako istu možemo unijeti preko administrativnog sučelja, zbog prirode ovih unosa, bilo je potrebno implementirati istu na korisničkom dijelu aplikacije.

Slika 11. Forma za unos mjesečne proizvodnje

5.3.6. Pregled poduzeća

Pregled stavki koje su važne za poduzeće nužan je za korisničko sučelje iako mu možemo pristupiti i uz pomoć administracijskog sučelja. Na slikama 12, 13 i 14 nalaze se pregledi zaposlenika, pogona i proizvoda unutar poduzeća (za potrebe simulacije). Pregledi su implementirani na način da su podatci ispisani direktno iz baze podataka za respektivnu tablicu, a prikazani su uz pomoć HTML listi.



Slika 12. Pregled zaposlenika



Slika 13. Pregled pogona



Slika 14. Pregled proizvoda

5.3.7. Izvještaji

Budući da se radi o simulaciji upravljačko-izvještajnog informacijskog sustava lako je zaključiti da su izvještaji najvažniji dio ove simulacije. Izvještaji nam služe kako bi relevantne podatke predložili na lako čitljiv i razumljiv način što znači da stvaramo informaciju koja je potrebna menadžmentu kako bi bio upućen u stanje proizvodnje te lakše donio poslovnu odluku. Pritiskom na neki od izvještaja, otvara se nova stranica s pregledom ispisa iz baze za respektivnu godinu. Kao što je ranije navedeno, Django koristi MVC arhitekturu te je potrebno implementirati svaki od dijelova arhitekture kako bi se ostvario rezultat. Django dolazi uz određene generične poglede koje je preporučljivo koristiti kako bi se izbjeglo implementiranje već implementiranih mogućnosti koje dolaze uz Django. Za implementaciju ovog izvještaja koristit ćemo jedan takav pogled, a to je ListView te budući da koristimo isti, potrebno je napraviti pogled baziran na klasi. Ovako to izgleda u datoteci views.py:

```
class Izvjestaj2016(ListView):
    model = Mjproizvodnja
    template_name = 'izvjestaj2016.html'
```

Klasa nosi naziv Izvjestaj2016, a unutar nje moramo deklarirati model s onim modelom na koji se odnosi pogled te zatim moramo deklarirati naziv template-a koji koristimo za prikaz istog. Nakon što je kreiran pogled, možemo napraviti template koji je zapravo stranica koja će se prikazati nakon što odaberemo neki od izvještaja. Ovako izgleda datoteka izvjestaj2016.html koja se nalazi u templates direktoriju:

```
{% extends 'base.html' %}

{% block content %}
<h2> Ukupni izvještaj za godinu 2016 </h2>

<div class="leftmenu">
<a href="/izvjestaj2016/chart2016/"> Grafički prikaz </a>
<h3> Izvještaji za Pogon1 </h3>
{% for mjproizvodnja in object_list %}
{% if mjproizvodnja.godina == 2016 and mjproizvodnja.pogon.naziv ==
'Pogon1' %}
    <div class="mjesec">
    <p> <b>Mjesec: </b>{{mjproizvodnja.mjesec}}</p>
    <ul>
    <li><p> <b>Proizvod:</b> {{ mjproizvodnja.proizvod}} </p></li>
    <li><p> Količina: {{ mjproizvodnja.kolicina}} </p></li>
    </ul>
    </div>
{% endif %}
{% endfor %}
</div>
```

```

<div class="rightmenu">
<br>
<h3> Izvještaji za Pogon2 </h3>
{% for mjproizvodnja in object_list %}
{% if mjproizvodnja.godina == 2016 and mjproizvodnja.pogon.naziv ==
'Pogon2' %}
    <div class="mjesec">
        <p> <b>Mjesec: </b>{{mjproizvodnja.mjesec}}</p>
        <ul>
            <li><p> <b>Proizvod:</b> {{ mjproizvodnja.proizvod}} </p></li>
            <li><p> Količina: {{ mjproizvodnja.kolicina}} </p></li>
        </ul>
    </div>
    {% endif %}
{% endfor %}
</div>
{% endblock %}

```

Može se uočiti kako je svaki izvještaj formiran u obliku HTML liste zbog lakše čitljivosti. Iznad liste se nalazi mjesec na koji se odnose podatci, a unutar liste su podatci vezani uz mjesečnu proizvodnju. Budući da su izvještaji odvojeni po pogonu, template je podijeljen na dva dijela te svaki ispisuje za po jedan pogon. Također je važno uočiti da se cijeli izvještaj nalazi unutar bloka koji proširuje base.html koji služi kao glavni template za prikaz stranica. Nakon što smo kreirali pogled i template koji će biti prikazan, nužno je dodati taj pogled s određenom putanjom u datoteku urls.py kako bi se pogled prikazao nakon što pritisnemo na hipervezu. Ovako to izgleda u datoteci urls.py:

```

urlpatterns = [
    ...
    path('izvjestaj2016/', Izvjestaj2016.as_view(), name='izvjestaj2016'),
    ...
]

```

Putanja za ovaj izvještaj je 'izvjestaj2016/', a budući da se radi o pogledu baziranom na klasi, važno je dodati `.as_view()` uz naziv pogleda jer on označava metodu koja se brine za pravilan prikaz pogleda na stranici. Da su korišteni pogledi bazirani na funkcijama, to ne bi bilo potrebno jer bi svaka funkcija vraćala response što je u ovom slučaju, template.

Ukupni izvještaj za godinu 2016

Grafički prikaz

Izvještaji za Pogon1

Mjesec: Siječanj

- **Proizvod:** Proizvod1
- **Količina:** 35

Mjesec: Veljača

- **Proizvod:** Proizvod1
- **Količina:** 35

Mjesec: Ožujak

- **Proizvod:** Proizvod1
- **Količina:** 38

Izvještaji za Pogon2

Mjesec: Siječanj

- **Proizvod:** Proizvod1
- **Količina:** 20

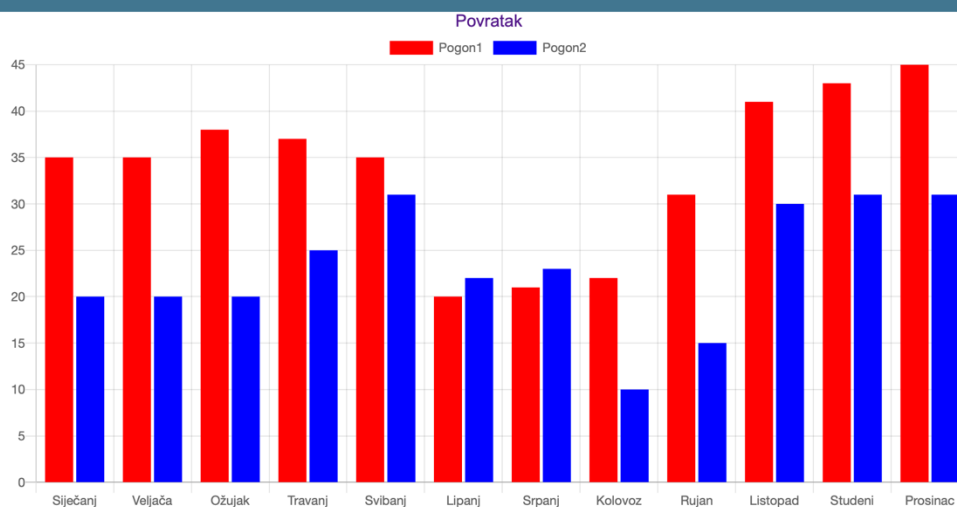
Mjesec: Veljača

- **Proizvod:** Proizvod1
- **Količina:** 20

Mjesec: Ožujak

- **Proizvod:** Proizvod1
- **Količina:** 20

Slika 15. Prikaz izvještaja za 2016. godinu



Slika 16. Grafički prikaz izvještaja za 2016. godinu

Na slici 15. možemo vidjeti kako izvještaj izgleda nakon što se otvori neka od stranica izvještaja koji su kreirani. Iznad izvještaja nalazi se mogućnost za prikaz grafičkog prikaza istog što je prikazano na slici 16. Grafički prikaz je ostvaren na način da je unutar template-a ubačen skriptni dio s JavaScript kodom koji crta graf prema zadanim podacima.

Dodatni izvještaji obuhvaćaju izvještaje vezane uz neispunjenje norme i udio ukupnih radnika prema broju radnika po svakom od pogona. Pri kreiranju izvještaja vezanog uz norme korištena je funkcija unutar modela pod nazivom razlika koju sam spomenuo nešto ranije.

[Administracija](#) / [Početna stranica](#)

Mjeseci u kojima norma nije ispunjena

[Grafički prikaz udjela po pogonu](#)

Pogon1 Kolovoz 2016

Razlika: 3

Pogon1 Rujan 2016

Razlika: 4

Pogon1 Kolovoz 2017

Razlika: 2

Pogon1 Rujan 2017

Razlika: 7

Pogon1 Srpanj 2018

Razlika: 10

Pogon1 Kolovoz 2018

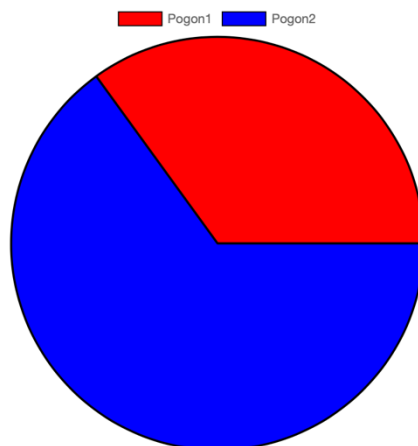
Razlika: 5

Slika 17. Izvještaj s neispunjenim normama

[Administracija](#) / [Početna stranica](#)

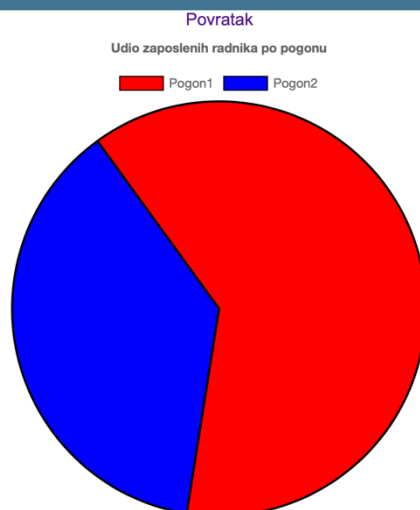
[Povratak](#)

Udio neispunjene norme prema pogonu



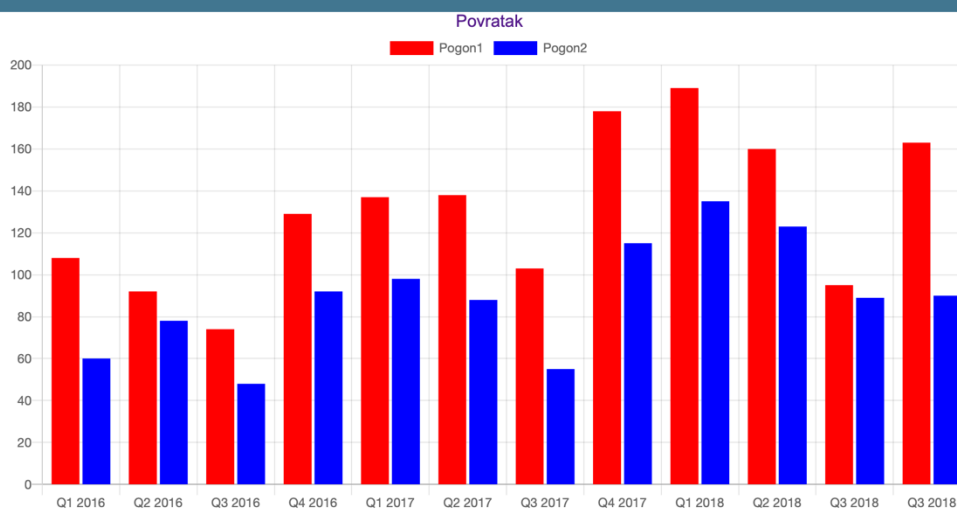
Slika 18. Grafički prikaz udjela po neispunjenju norme

Na slici 17 se vidi rezultat kreiranog izvještaja o neispunjenim normama. Označen je pogon, mjesec i godina u kojoj nije ostvarena norma te razlika između ostvarene proizvodnje i zadane norme. Slika 18 pokazuje grafički prikaz udjela po pogonima koji nisu ostvarili normu kako bi se lakše utvrdilo koji pogon ima više problema s proizvodnjom.



Slika 19. Grafički prikaz udjela radnika po pogonu

Na slici 19 možemo vidjeti grafički prikaz udjela radnika po pogonu što je poprilično korisno u ovoj situaciji budući da je jasno da jedan od pogona ima poprilično lošiji rezultat proizvodnje od drugog pogona.



Slika 20. Grafički prikaz ukupne proizvodnje za razdoblje 2016-2018

Slika 20 prikazuje ukupnu proizvodnju u posljednje tri godine te budući da bi prikaz sa svih 36 mjeseci bio nepregledan, proizvodnja je prikazana u kvartalima koji su dovoljno slični da bi se mogle primijetiti neželjeni rezultati te slabija proizvodnja u nekom od kvartala. Ovo je najvažniji izvještaj za strateško upravljanje.

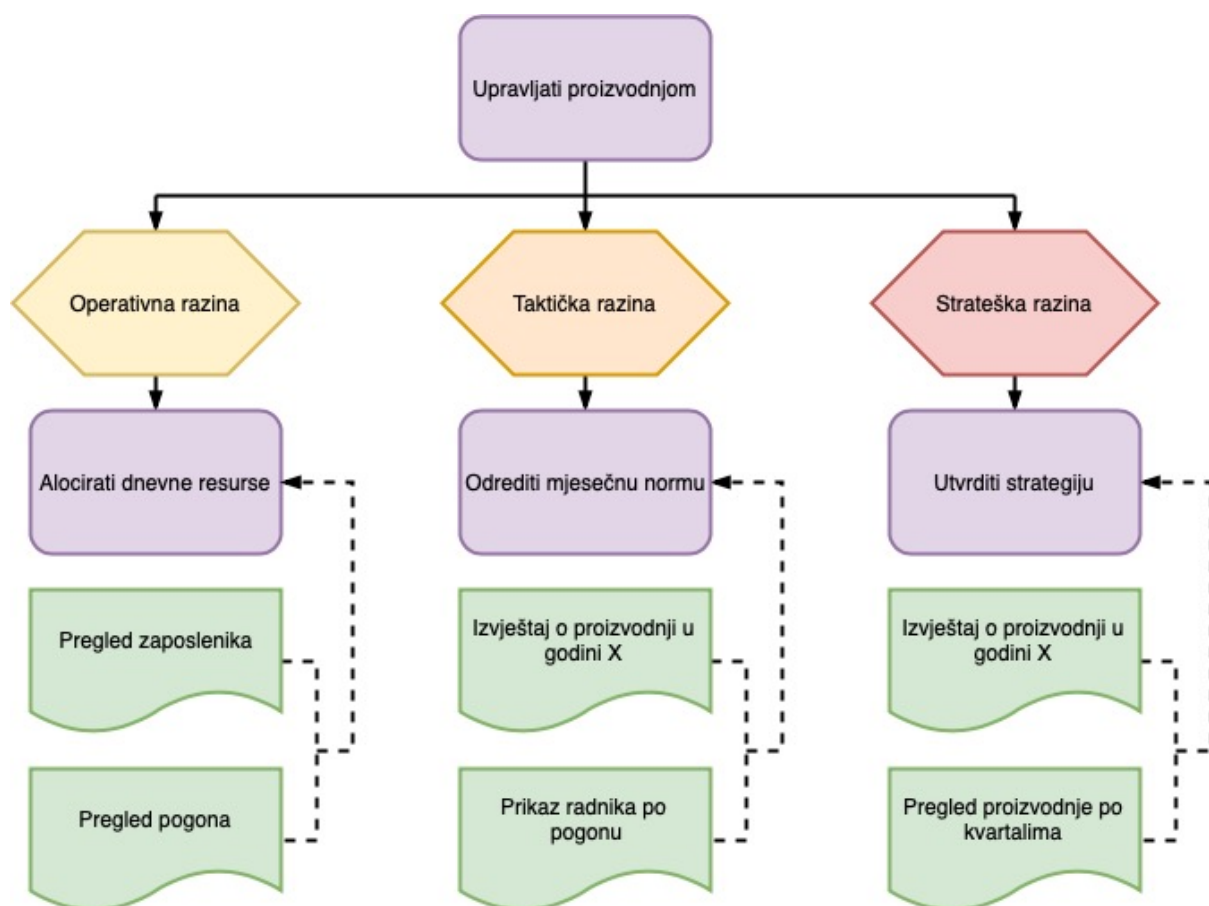
5.4. Procesi na upravljačkoj razini simulacije

Slika 21 prikazuje dijagram s procesima koji se odvijaju na upravljačkoj razini simulacije zajedno s izvještajima koji stvaraju informacije za izvođenje istih. Operativna razina zadužena je za automatizirane dnevne i tjedne odluke te je ona najniža razina upravljanja. U slučaju simulacije, operativna razina je zadužena za alociranje radnih resursa, a pri tome joj pomažu pregled radnika i pogona, iako nisu klasični izvještaji. Taktička razina je zadužena za odluke koje nisu potpuno automatizirane i na očigled jasne, stoga taktička razina pripada srednjem menadžmentu i u slučaju simulacije, zadužena je za određivanje mjesečne norme budući da zbog nepredvidivih okolnosti, isto nije moguće uvijek jasno odrediti te joj u tome pomažu informacije iz izvještaja o proizvodnji u određenoj godini te naravno grafički prikaz radnika po pogonu. Strateška razina je ona najviša te je ona zadužena za samu strategiju poduzeća, ali u slučaju simulacije, ona utvrđuje strategiju vezanu uz proizvodnju budući da je simulacija provedena na izoliranom dijelu sustava. Pri donošenju strateških odluka pomažu informacije iz izvještaja o proizvodnji u određenoj godini te informacije s pregleda proizvodnje po kvartalima. Na slici su izvori informacija označeni zelenom bojom.

Pregled zaposlenika i pogona je nužan za operativnu razinu prilikom izvođenja procesa Alocirati dnevne resurse budući da ti pregledi nude informacije o zaposlenicima i raspoloživim pogonima kako bi za to zadužene osobe mogle lakše provoditi navedeni proces. Na svakom od pregleda nalaze se informacije o entitetima koji su vezani uz zaposlenike i pogone.

Izvještaj o proizvodnji u nekoj od godina i grafički prikaz radnika po pogonu pružaju informacije za lakše provođenje procesa Odrediti mjesečnu normu, koji se odvija na taktičkoj razini. Određivanje mjesečne norme temelji na aproksimaciji moguće vrijednosti koja je dobivena uz pomoć informacija o prijašnjoj proizvodnji za određeni mjesec. Grafički prikaz radnika po pogonu nam daje na uvid udio zaposlenika prema svakom pogonu što pomaže pri određivanju norme budući da nas broj dostupnih radnika u pogonu itekako ograničava prilikom izračuna željene norme.

Izvještaj o proizvodnji za neku od godina također se koristi pri donošenju poslovnih odluka na strateškoj razini budući da grafički prikaz navedenog izvještaja daje informaciju o kretanju proizvodnje po mjesecima koja je nužna kako bi bilo moguće utvrditi uzorak tijekom godina, npr. manja proizvodnja u ljetnim mjesecima zbog godišnjih odmora. Pregled proizvodnje po kvartalima nudi sličnu informaciju, ali u kvartalima kako bi se uspostavilo ima li napretka nakon određenog broja godina po pitanju zadovoljavanja proizvodnje, te utvrđivanja strategije proizvodnog poduzeća.



Slika 21. Prikaz procesa na upravljačkoj razini

6. Zaključak

Informacijski sustavi su svugdje oko nas i teško je ne primijetiti njihov utjecaj na okolinu i društvo. Poslovni informacijski sustavi su vrlo važan faktor u poslovanju modernih poduzeća i trenutno je nemoguće zamisliti poslovanje bez njih te su oni zaslužni za bolje savladavanje poslovnih prepreka te za opskrbu relevantnim informacijama koje čine veliku razliku ukoliko ih dobijemo na vrijeme. Postoji više vrsta poslovnih informacijskih sustava, ali upravljački informacijski sustav, iako hijerarhijski nije previsoko, jedan je od najvažnijih informacijskih sustava budući da je upravo on zadužen za opskrbu pravovremenim i relevantnim informacijama koje su nužne za donošenje poslovnih odluka. Upravljački informacijski sustavi imaju veliki utjecaj na smjer prema kojem će poduzeće težiti te je zbog toga važno implementirati isti na način koji je prikladan za određeno poduzeće.

Simulacija upravljačko-izvještajnog informacijskog sustava implementirana je uz pomoć web tehnologija od kojih je najvažnija Django, web razvojni okvir napisan u programskom jeziku Python. Odlučio sam se za web tehnologije budući da su one uvijek korisne za znati, a Django sam po sebi je trenutno jedan od najpoznatijih razvojnih okvira budući da ga koriste neke od svjetski poznatih kompanija za svoja softverska rješenja. Simulacija iako nije preopširna, prikazuje glavnu srž upravljačko-izvještajnog informacijskog sustava, a to su izvještaji. Prikazano je više izvještaja, a za jedan od njih je prikazan i način na koji je isti implementiran.

Zadovoljan sam odabirom teme budući da je poprilično dobra kao zaključak mog trogodišnjeg studija informatike te sam se potrudio primijeniti naučena teorijska i praktična znanja u izradi ovog rada. Znanja stečena tijekom studija će uvelike značiti u mom daljnjem napretku, a pogotovo znanja vezana uz informacijske sustave.

Popis literature

1. Pejić Bach, M., Varga, M., Srića, V., Spremić, M., Bosilj Vukšić, V., Ćurko, K., ... Jaković, B. (2016.), *Informacijski sustavi u poslovanju*, Zagreb: Ekonomski fakultet – Zagreb
2. Stair, R., Reynolds, G. (2010), *Principles of information systems, a Managerial approach (ninth edition)*, Boston, SAD: Course Technology, Cengage Learning
3. Panian, Ž., Ćurko, K., Bosilj Vukšić, V., Čerić, V., Pejić Bach, M., Požgaj, Ž., ... Varga, M. (2010.), *Poslovni informacijski sustavi*, Element Zagreb, Zagreb
4. Leksikografski zavod Miroslav Krleža (2019.), Hrvatska enciklopedija, Zagreb, dostupno na: <http://www.enciklopedija.hr/natuknica.aspx?id=56068>
5. Python Software Foundation (2019), Python Language Reference, verzija 3.7.4 dostupno na: <https://www.python.org>
6. Django Software Foundation (2019), Django (verzija 2.2.3), dostupno na: <https://www.djangoproject.com>

Popis slika

Slika 1. Funkcije poslovnog sustava	6
Slika 2. Primjer ER dijagrama.....	8
Slika 3. Slojevi poslovnog informacijskog sustava.....	9
Slika 4. Transakcijski proces.....	10
Slika 5. Prikaz promatranog procesa.....	14
Slika 6. Prikaz upravljačkog sustava.....	15
Slika 7. ER model simulacije	17
Slika 8. Administracija simulacije	21
Slika 9. Forma za dodavanje proizvoda	21
Slika 10. Prikaz početne stranice.....	22
Slika 11. Forma za unos mjesečne proizvodnje	22
Slika 12. Pregled zaposlenika.....	23
Slika 13. Pregled pogona.....	23
Slika 14. Pregled proizvoda	23
Slika 15. Prikaz izvještaja za 2016. godinu.....	26
Slika 16. Grafički prikaz izvještaja za 2016. godinu.....	26
Slika 17. Izvještaj s neispunjenim normama	27
Slika 18. Grafički prikaz udjela po neispunjenju norme	27
Slika 19. Grafički prikaz udjela radnika po pogonu.....	28
Slika 20. Grafički prikaz ukupne proizvodnje za razdoblje 2016-2018.....	28
Slika 21. Prikaz procesa na upravljačkoj razini	30