

# Izrada poslovne web aplikacije u Oracle APEX 20.1

---

Štorić, Luka

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:285318>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku

Diplomski studij Informatike

Poslovna informatika

Luka Štorić

Izrada poslovne web aplikacije u Oracle

APEX 20.1

Diplomski rad

Mentor: prof. dr. sc. Patrizia Pošćić

Rijeka, 13. srpnja 2021.

## Sadržaj

1. Zadatak.....	1
2. Sažetak i ključne riječi .....	2
3. Uvod.....	3
4. Proces izrade web aplikacije.....	4
4.1. Izrada relacijskog modela .....	4
4.1.1. Dijagram Entiteti-Veze i poslovni sustav – pojašnjenje .....	4
4.1.2. Relacijska shema.....	9
4.2. Izrada nove aplikacije .....	10
4.3. Izrada tablica .....	12
4.4. Izrada korisničkog računa .....	15
4.5. Izrada stranica aplikacije.....	17
4.5.1. Stranice Popis i Unos - kreiranje .....	17
4.5.2. Stranice Popis i Unos – uređivanje .....	19
4.5.3. Stranica Popis Evidencijskih Listova (Administrator / Dostavljač) .....	22
4.5.4. Stranica Unos Evidencijskog Lista .....	26
4.5.5. Stranica Unos Dostava .....	27
4.5.6. Stranica Popis Osnovica Dostavljača.....	27
4.5.7. Stranica Unos Osnovice Dostavljača .....	28
4.5.8. Stranica Popis Plaća Dostavljača .....	28
4.5.9. Stranica Unos Plaće Dostavljača .....	29
4.5.10. Stranica Metrike.....	30
4.5.11. Početna stranica (Administrator / Dostavljač) .....	32
4.5.12. Stranice O Aplikaciji i <i>Feedback</i> .....	35
4.5.13. Stranica Login.....	36
4.6. Grafički dizajn stranica .....	37
5. Usporedba alata Oracle APEX i SoftVelocity Clarion .....	38
5.1. SoftVelocity Clarion .....	38
5.2. Oracle APEX.....	39
6. Zaključak.....	40
Literatura.....	41
Popis slika .....	42
Prilozi .....	43
Zadatak za Diplomski rad .....	43

# 1. Zadatak

Zadatak ovog rada je osmisliti i kreirati poslovnu web aplikaciju nad relacijskom bazom podataka za dostavnu službu. Odabrani alat za izradu je Oracle Application Express (APEX) 20.1. Uz to potrebno je pismeno detaljno opisati sam proces izrade te potkrijepiti aplikaciju teorijskom pozadinom. Također, potrebno je napraviti kratku usporedbu alata APEX i Clarion, obzirom da je u tom alatu kreirana *desktop* aplikacija na kojoj će se temeljiti ovaj rad.

## **2. Sažetak i ključne riječi**

Rad je koncipiran na način da prvo daje teorijsku pozadinu relacijskih baza podataka. Zatim, koristeći se danom teorijom, objašnjava kreiranje relacijske baze podataka za odabrani poslovni sustav. Nakon kreiranja Dijagrama Entiteti-Veze te relacijskog modela i sheme, prelazi se na kreiranje same aplikacije. Dan je detaljan uvid u korake samog procesa, od kreiranja Radnog prostora do svake stranice posebno. Finalno, uspoređeni su alati APEX i Clarion te su dane njihove najbitnije sličnosti i razlike.

Ključne riječi: Aplikacija, Relacija, Baza podataka, Relacijska shema, Relacijski model, Dijagram Entiteti Veze, Oracle, APEX, Tablice, Veze, Entiteti, Agregacija, SQL, JavaScript, Web, Server

### 3. Uvod

Poslovna aplikacija nad relacijskom bazom podataka danas je neizostavan element svakog poslovanja. Bila njezina uloga vođenje poslovnih izvještaja, upravljanje ljudskim resursima te imovinom ili nešto treće, glavna zadaća ostaje ista: povezati korisnika s pripadajućom bazom podataka.

Današnji je svijet umrežen internetom u gotovo svakom aspektu. Tako su i poslovne aplikacije, s dosadašnjih *desktop* verzija, većinom prešle na web oblik. Na taj način su vrlo lako dostupne svim potrebnim korisnicima jer oni ne moraju imati datoteke aplikacije kao ni samu bazu na fizičkom uređaju. Umjesto toga, baza se nalazi na udaljenom serveru koji zauzvrat nudi visoku stopu sigurnosti.

U ovom radu prikazan je jedan takav prelazak. Aplikacija koja će se izraditi zapravo je nadogradnja postojeće *desktop* aplikacije za dostavnu službu istog autora [1]. Kao alat korišten je sustav Application Express (APEX) američke multinacionalne kompanije Oracle. APEX kroz intuitivno sučelje omogućuje korisniku široku lepezu mogućnosti, kako logičkih i pozadinskih funkcija same aplikacije tako i *front-end* komponenti.

Prvi dio rada opisuje, kroz teorijsku pozadinu, zadanu bazu podataka. Prikazat će se Dijagram Entiteti-Veze koji predočuje samu bazu, te će se detaljno rastumačiti svi pripadni elementi.

U drugom dijelu opisan je proces kreiranja tablica u Oracle-ovoj bazi, potrebnih veza odnosno ključeva, te postavljanje korisničkih računa koji će poslije služiti za prijavu u samu aplikaciju.

Treći dio rada biti će najopširniji. Ovdje će se detaljno prikazati izrada samih stranica u aplikaciji, dodavanje potrebnih elemenata te korištenje velikog broja mogućnosti i opcija koje APEX nudi, a sve popraćeno *screenshot*-ovima samog sučelja za bolje razumijevanje procesa izrade aplikacije.

Konačno, prije zaključka, usporedit će se dva različita RAD (*Rapid Application Development*) alata korištena pri izradi odabranih aplikacija. Radi se o ovdje korištenom APEX-u te softveru Clarion američke tvrtke SoftVelocity čija je primarna namjena kreiranje *desktop* aplikacija. Ukazat će se na njihove očigledne razlike, ali i manje primjetne sličnosti kao i prednosti i nedostatke svakog zasebnog alata.

Aplikacija čiju izradu opisuje ovaj rad nazvana je DeliServ Web što pokazuje oblik na koji je postojeća aplikacija nadograđena.

## 4. Proces izrade web aplikacije

### 4.1. Izrada relacijskog modela

Kod izrade svake aplikacije osnovni preduvjet je postojeći relacijski model. Radi se o temeljnom opisu poslovnog procesa s detaljnim prikazom osnovnih elemenata i pripadnih odnosa.

Pojam relacijskog modela nastao je 1960-tih godina, a autorom se smatra Edgar Codd. Ovako ga on opisuje: „Neka su dani skupovi  $D_1, D_2, \dots, D_n$  (ne obavezno različiti),  $R$  je relacija (engl. *Relation*) nad ovih  $n$  skupova ( $n > 0$ ) ako je to skup  $n$ -torke takav da za svaku  $n$ -torku vrijedi da je prvi element  $n$ -torke iz  $D_1$ , drugi iz  $D_2, \dots$ ,  $n$ -ti iz  $D_n$ . Definiciju se može lakše shvatiti iz zapisa relacije u obliku *Naziv\_relacije* (*Naziv\_atributa1, Naziv\_atributa2, ...*) tj. ako je neposredno zapišemo kao *Naziv\_tablice* (*Naziv\_stupca1, Naziv\_stupca2...*). Zapis svih tih relacija iz modela naziva se relacijska shema baze podataka.“ [2].

Osim relacijskog modela, koristi se i model Entiteti-Veze iz razloga što daje dobar grafički prikaz same relacijske sheme. O modelu EV referenca [3] govori da se radi o semantički bogatoj metodi za modeliranje podataka jer raspolaže ljudski bliskim konceptima. Modelu pripadni dijagram je „... grafički prikaz modela podataka sustava, metodom EV“ [3]. Upravo taj dijagram (DEV) bit će polazna točka prikaza odabrane sheme. Osnovi pojmovi DEV-a (TIP ENTITETA, VEZA, ATRIBUT, SLAB TIP ENTITETA i SPECIJALNE VEZE te AGREGACIJA) biti će prikazani i objašnjeni u nastavku, a sve potkrijepljeno značenjem odabranog sustava. Na osnovu modela DEV kreira se relacijski model.

Bitno je naglasiti da je ovaj poslovni proces kombinacija stvarnih procesa, različitih primjera kao i autorove imaginacije. Literatura [3] poslužiti će za opis i objašnjenje svakog elementa kao i međusobne interakcije.

#### 4.1.1. Dijagram Entiteti-Veze i poslovni sustav – pojašnjenje

Da bi se ovakav dijagram shvatio, potrebno je moći razumjeti njegove elemente kao i njihov odnos. U nastavku je objašnjenje glavnih koncepata DEV-a.

Tip entiteta je osnovni koncept. On je „... neki njegov realni ili konceptualni element, to je neka posebnost što u poslovnome sustavu postoji i jasno se razlikuje od drugih entiteta.“ [3]. Prikazani kao pravokutnici s veliko tiskanim nazivima zapravo predstavljaju tablice baze podataka.

Svaka tablica sastoji se od stupaca. Stupci su atributi tipa entiteta. Atribut je svojstvo entiteta koje na Dijagramu predstavlja simbol elipse s malo tiskanim imenima.

Dva uvjeta određuju je li neki atribut (ili više njih) ključ relacije. Radi se o Uvjetu jedinstvenosti<sup>1</sup> i Uvjetu neredundantnosti<sup>2</sup>. U slučaju kada doista više atributa ispunjava ove

---

<sup>1</sup> „Vrijednost ključa svake  $n$ -torke relacije jedinstveno određuje  $n$ -torku, to jest, ne postoje dva takva retka u tablici da imaju sve iste vrijednosti svih atributa koji čine ključ.“ [3]

<sup>2</sup> „Ne postoji niti jedan atribut kao dio ključa koji se može izostaviti iz ključa, a da se pritom uvjet jedinstvenosti ne gubi, to jest, ključ je unija minimalnoga broja atributa.“ [3]

uvjete, oni čine kandidate za ključ. Potrebno je odabrati jedan ključ koji će biti primarni ključ relacije. Za njega je bitno da „...je jednoj vrijednosti atributa u relaciji pridružena samo jedna vrijednost ostalih atributa relacije, odnosno da su svi neključni atributi funkcijski ovisni o ključu.“[3]. Primarni ključ označava se podcrtavanjem punom linijom.

Atribut koji nije ključ relacije kojoj pripada, ali je ključ u nekoj drugoj relaciji tog modela, naziva se vanjski ključ a podcrtava se isprekidanom linijom.

Slabi tip entiteta još je jedan od koncepata relacijskog modela. Literatura [3] ga opisuje: „Slabi tip entiteta (engl. *weak entity*) je tip entiteta koji je na neki način ovisan o nekom drugome tipu entiteta, a ta se ovisnost prikazuje specijalnim tipom među jakim i slabim tipom entiteta.“. Ključ takvog tipa entiteta sastavljen je od ključa nadtipa i ključa podtipa.

Konačno, bitno je objasniti i agregaciju. Agregirani tip entiteta zapravo je „...apstrakcija u kojoj se tip veze između dvaju ili više tipova tretira kao novi tip entiteta.“[3]. Ključ agregacije sastoji se od ključeva tipova entiteta koje povezuje. Za agregaciju su bitne brojnosti veze o kojima će kasnije biti riječi.

Na ovom mjestu potrebno je objasniti i međusobne odnose različitih tipova entiteta. Veze koje povezuju tipove entiteta definirane su kao „koncept koji predstavlja neku interakciju među tipovima entiteta u sustavu, odnosno predstavlja znanje o njihovoj povezanosti.“[3].

Svaka veza ima jasno definiranu brojnost. „Brojnost tipa veze je broj koji kaže koliko entiteta pojedinoga tipa entiteta  $E_1$  sudjeluje (pojavljuje se) u tipu veze  $V$  s entitetom iz tipa entiteta  $E$ .“[3]. Taj broj se prikazuje jedino na DEV-u, u obliku uređenog para brojeva. Prvi broj označava Donju granicu, a drugi Gornju. Pozicija na koju se upisuje brojnost je na samoj vezi, što bliže pripadnom tipu entiteta.

Uz objašnjene glavne pojmove, lakše je shvatiti DEV odabranog sustava.

Krenimo od tipa entiteta. Iako je DOSTAVA, po osjećaju, centralni tip entiteta, započet ćemo od tipa entiteta EVIDENCIJSKI\_LIST. On predstavlja zapis dokumenata koji sadrže detaljne informacije o dostavama. Primarni ključ je *Sifra\_ev\_lista* kao jedinstveni identifikator takvog dokumenta. Ostali atributi su *Datum\_ev\_lista* te *Broj\_dostava\_u\_danu*. Ovdje se računa ukupan broj dostava koje su zapisane na pojedinom Evidencijskom listu. Još jedan pripadni atribut je *Zaključano* koji označava da je list zaključen i promjene nisu dozvoljene. Ovaj tip entiteta sadrži i attribute koji su u službi vanjskog ključa. One se po pravilu crtanja dijagrama EV ne zapisuju na ovom mjestu. Prvi takav atribut je *Sifra\_dost\_centra*. To je primarni ključ tipa entiteta DOSTAVNI\_CENTAR.

U ovoj se tablici zapisuju podaci o dostavnim centrima koje posjeduje ova služba. Atributi tipa entiteta, uz navedeni primarni ključ, su *Naziv\_dost\_centra*, koji je najčešće prema najbližem gradu ili mjestu, *Adresa\_dost\_centra*, *Podrucje\_djelovanja* kao podatak o točnom okrugu u kojem taj centar ima jurisdikciju, *Broj\_dostavljača* te *Broj\_vozila* kao podaci o dostavljačima i vozilima koji su dodijeljeni ovom centru. Tablica DOSTAVNI\_CENTAR povezana je vezom tipa (1,1)(0,M) s tablicom EVIDENCIJSKI\_LIST. Veza s tom brojnošću čita se na način: „Jedan dostavni centar može se nalaziti na najmanje jednom, a najviše mnogo evidencijskih listova. Jedan evidencijski list može sadržavati najmanje jedan, ali i najviše jedan dostavni centar.“ Ovo je specijalan tip veze, kod kojeg nalazimo vanjski ključ te time gore navedeno pravilo o pisanju.

Sljedeći vanjski ključ tipa entiteta EVIDENCIJSKI\_LIST je *Broj\_dost\_vozila*.

U tipu entiteta DOSTAVNO\_VOZILO nalazimo atribut i primarni ključ istog imena. Radi se o jedinstvenom broju dodijeljenom svakom vozilu koji se evidentira kroz ovaj stupac tablice.



Dalje, imamo attribute *Marka*, kao zapis o proizvođaču vozila, *Model*, *Godina* (proizvodnje), *Datum\_regist* kao datum posljednje registracije te *Datum\_servisa* kao datum zadnjeg servisa na vozilu. I ovdje nalazimo vezu (1,1)(0,M) te tako i pripadno pravilo o pisanju ključa. Posljednji atribut je *Kvar* kao zastavica je li auto pokvaren ili ne. Svrha ove tablice je evidencija vozila koje tvrtka posjeduje.

Zadnji vanjski ključ koji se pojavljuje u tipu entiteta EVIDENCIJSKI\_LIST je *Sifra\_dostavljacka*.

Sada znamo da je veza s tipom entiteta DOSTAVLJAC, u tom slučaju, sigurno tipa (1,1)(0,M). Naravno dostavna služba ne može postojati bez dostavljača, pa je ovo još jedna prirodno prisutna tablica. Atributi ovog tipa entiteta, uz primarni ključ, su, naravno, *Ime\_dostavljacka* i *Prezime\_dostavljacka*, *Kontakt* (broj telefona), *Status\_dostavljacka* (u smislu je li aktivan, trenutno na godišnjem odmoru, bolovanju itd.) te *Username* kao korisničko ime koje će biti korišteno za prijavu u aplikaciju.

Tip entitet DOSTAVLJAC povezan je s tipom entiteta OSNOVICA preko agregacije OSNOVICA\_DOSTAVLJACA. Na taj način dobije se osnovica svakog zasebnog dostavljača. Atributi tipa entiteta OSNOVICA su *Rb\_osnovice* (primarni ključ) i *Iznos\_osnovice*. Agregacija OSNOVICA\_DOSTAVLJACA ima samo pripadni primarni ključ (sastavljen od ključeva *Sifra\_dostavljacka* i *Rb\_osnovice*) odnosno nema ostalih atributa.

Na sličan način, povezani su tipovi entiteta (agregacija) OSNOVICA\_DOSTAVLJACA i BONUS. Povezuje ih agregacija PLACA\_DOSTAVLJACA. Kao što i imena govore, povezivanjem osnovice svakog zasebnog dostavljača i dodijeljenog bonusa dobije se ukupna plaća dostavljača. Tipovi entiteta BONUS sadrži attribute *Rb\_bonusa*(PK) i *Iznos\_bonusa*. Agregacija PLACA\_DOSTAVLJACA sastoji se od primarnih ključeva tipa entiteta koje povezuje te atributa *Placa\_ukupno* u koju se nakon izračuna sprema iznos plaće za određenog dostavljača.

Sada možemo preći na tip entiteta DOSTAVA. Na dijagramu EV primjećujemo dupli četverokut. To je oznaka slabog tipa entiteta tj. prema tome znamo da uklanjanjem tipa entiteta EVIDENCIJSKI\_LIST, automatski se uklanja i DOSTAVA, jer je zapis o dostavama ovisan o zapisu evidencijskih listova. To nam pokazuje strelica, umjesto obične oznake veze. U ovom slučaju, također se ne upisuju brojnosti na dijagram. Atributi ovog slabog tipa entiteta su *Sifra\_dostave*, *Tezina* (težina pošiljke koja će se koristiti pri izračunu cijene), *Vrijednost\_u\_kn* (Potrebno pri odabiru usluge, razine čuvanja itd. Biti će detaljnije objašnjeno kod tipa entiteta USLUGA) te *Cijena\_dostave\_ukupno* (stupac u koji će se zapisati rezultat izračuna troška dostave. Izračun će biti u smislu da se množe težina predmeta i cijena po gramu definirana u tablici TIP te se tom umnošku zbraja cijena usluge iz tablice USLUGA).

I ovdje nalazimo vanjske ključeve tj. attribute koji se po pravilu neće zapisati pod ovaj tip entiteta. Sukladno tome, može se zaključiti da je veza sa sljedećih 5 tipova entiteta tipa (1,1)(1,M). Svaki tip entiteta i veza biti će posebno razjašnjeni.

Prvi atribut, a ujedno i vanjski ključ je *OIB\_primatelja*. To je naziv atributa i primarnog ključa tablice PRIMATELJ. Ona zapisuje podatke o primatelju kojem je dostava upućena. Uz standardne attribute (*Ime\_primatelja*, *Prezime\_primatelja*, *Adresa\_primatelja* te *Kontakt*) nalazimo i atribut *Crna\_lista*. Prema svom zapisu, on će korisniku reći je li određeni primatelj stavljen na crnu listu tj. je li prema njemu obustavljena dostava.

Na isti način (isti atributi) kreiran je i tip entiteta POSILJATELJ koji daje informaciju o pošiljatelju koji je uputio dostavu.

Sljedeći vanjski ključ tablice DOSTAVA je *Sifra\_tipa*. On je povezuje s tablicom TIP. Radi se o zapisima raznih tipova dostave (pismo, paket...). Zato su atributi, uz primarni ključ, redom *Naziv\_tipa*, te *Cijena\_po\_g*. Ovaj zadnji atribut važan je u izračunu cijene dostave jer se upravo on množi sa težinom dostavljanog predmeta.

Dalje imamo atribut *Sifra\_usluge*, primarni ključ tablice USLUGA. Sve dostave nisu jednake. Ovisno o vrijednosti, hitnoći ili nekom drugom faktoru, pošiljatelj može izabrati uslugu koja mu se čini potrebnom. Atributi ovog tipa entiteta su: *Naziv\_usluge*, *Opis\_usluge*, *Razina\_sigurnosti* (brojeva vrijednost koja pokazuje službi prioritet u dostavi), *Garantirani\_broj\_dana* (zamišljeno je da što je odabrana usluga skuplja, obrnuto proporcionalno se smanjuje broj dana u kojima dostava mora stići do odredišta), *Cijena\_usluge* (iznos koji se dodaje u izračunu troška dostave) te *Minimalni\_potrebni\_iznos* (govori pošiljatelju koliki mora biti minimalni iznos vrijednosti da bi se ova usluga mogla iskoristiti).

Zatim slijedi vanjski ključ *Sifra\_stanja*. Radi se o primarnom ključu tablice STANJE\_POSILJKE. Ova tablica, uz atribut *Sifra\_stanja*, ima samo još jedan atribut, a to je *Opis\_stanja*. Svrha ove tablice je zamišljena da sadrži zapise o mogućim stanjima pošiljke (originalno, oštećeno, izgubljeno...). Kako služba svakog dana radi sa velikim brojem dostava, ove su situacije moguće pa je potrebno imati i zapis o njima.

Posljednji vanjski ključ tablice DOSTAVA je *Sifra\_statusa*. On dolazi iz tablice STATUS\_DOSTAVE. Svrha ove tablice je zapis o statusima dostave tj. je li ona u pripremi, u tranzitu, dostavljena... Uz atribut *Sifra\_statusa* tu je i *Opis\_statusa*.

Time je pojašnjen cijeli Dijagram Entiteti-Veze odabranog poslovnog sustava. Slika 1 prikazuje kreirani DEV.



#### 4.1.2. Relacijska shema

Da bi se shvatio pojam relacijske sheme, potrebno je objasniti što je zapravo relacijska baza podataka. Iz literature[3] saznajemo: „Relacijska baza podataka (engl. *relational database*) je skup u vremenu promjenljivih relacija opisanih u shemi baze podataka.“

Iz definicije strukture relacijskog modela saznajemo o shemi baze: „Struktura relacijskog modela definira se shemom relacijskog modela podataka. Elementi strukture relacijske sheme su relacije.“[3].

U nastavku je prikazana relacijska shema odabrane baze podataka dobivena na temelju modela Entiteti-Veze.

DOSTAVNI\_CENTAR (Sifra\_dost\_centra, Naziv\_dost\_centra, Adresa\_dost\_centra, Podrucje\_djelovanja, Broj\_dostavljacka, Broj\_vozila)

DOSTAVNO\_VOZILO (Broj\_dost\_vozila, Marka, Model, Godina, Datum\_regist, Datum\_servisa, Kvar)

DOSTAVLJAC (Sifra\_dostavljacka, Ime\_dostavljacka, Prezime\_dostavljacka, Kontakt, Status\_dostavljacka, Username)

OSNOVICA (Rb\_osnovice, Iznos\_osnovice)

BONUS (Rb\_bonusa, Iznos\_bonusa)

OSNOVICA\_DOSTAVLJACA (Sifra\_dostavljacka, Rb\_osnovice)

PLACA\_DOSTAVLJACA (Sifra\_dostavljacka, Rb\_osnovice, Rb\_bonusa, Placa\_ukupno)

STANJE\_POSILJKE (Sifra\_stanja, Opis\_stanja)

STATUS\_POSILJKE (Sifra\_statusa, Opis\_statusa)

PRIMATELJ (OIB\_primatelja, Ime\_primatelja, Prezime\_primatelja, Adresa\_primatelja, Kontakt\_primatelja, Crna\_lista\_Prim)

POSILJATELJ (OIB\_posiljatelja, Ime\_posiljatelja, Prezime\_posiljatelja, Adresa\_posiljatelja, Kontakt\_posiljatelja, Crna\_lista\_Pos)

TIP (Sifra\_tipa, Naziv\_tipa, Cijena\_po\_g)

USLUGA (Sifra\_usluge, Naziv\_usluge, Opis\_usluge, Cijena\_usluge, Razina\_sigurnosti, Garantirani\_broj\_dana, Minimalni\_potrebni\_iznos)

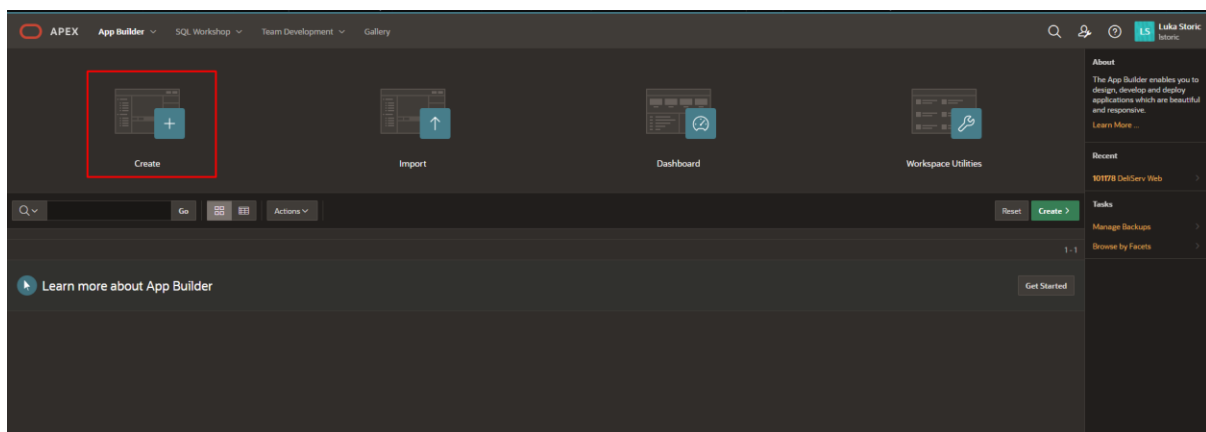
DOSTAVA (Sifra\_ev\_lista, Sifra\_dostave, OIB\_posiljatelja, OIB\_primatelja, Tezina, Vrijednost\_u\_kn, Cijena\_dostave\_ukupno, Sifra\_tipa, Sifra\_usluge, Sifra\_stanja, Sifra\_statusa)

EVIDENCIJSKI\_LIST (Sifra\_ev\_lista, Datum\_ev\_lista, Broj\_dostava\_u\_danu, Zakljucano, Sifra\_dost\_centra, Sifra\_dostavljacka, Broj\_dost\_vozila)

## 4.2. Izrada nove aplikacije

Oracle APEX prije upotrebe traži ispunjavanje zahtjeva za Radnim prostorom (*workspace*) kojim se provjerava stvarna namjena budućih aplikacija. U ovom slučaju odabrano je za potrebe studentskih obaveza i zahtjev se odobrava gotovo trenutačno.

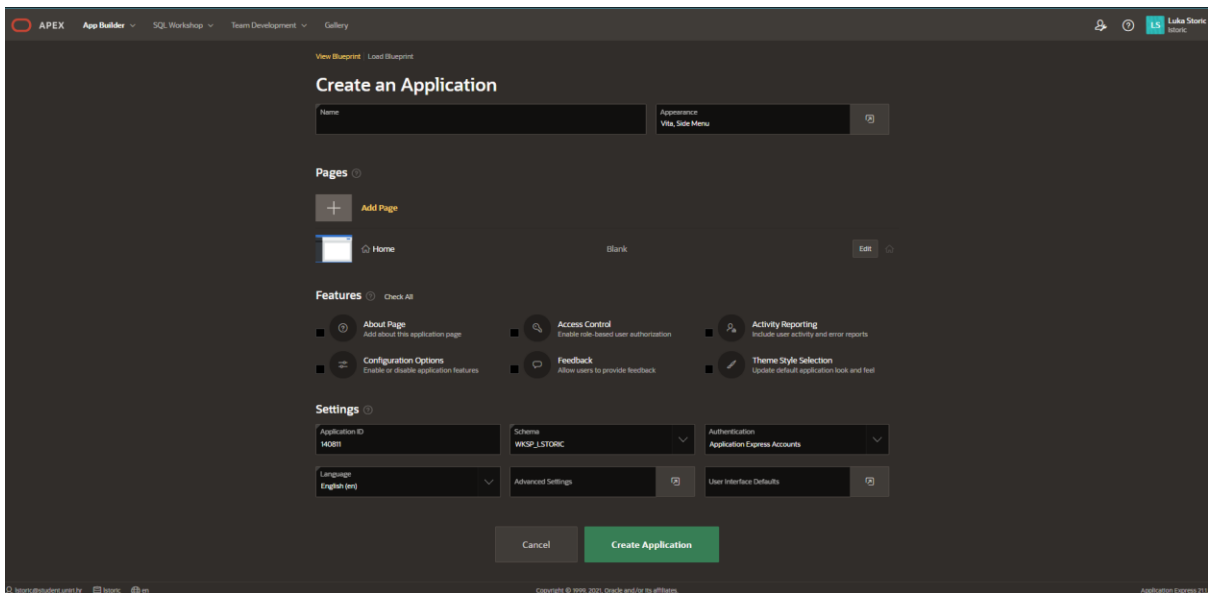
Po prijavi u APEX, odabiremo gumb *App Builder*. Tu se prikazuju sve aplikacije kreirane na ovom *workspace*-u. Dodatno, moguće je uvesti već kreirane aplikacije, pogledati podatke o samom *workspace*-u te urediti postavke. Odabiremo gumb *Create* kao što je prikazano na slici 2.



Slika 2. *App Builder*

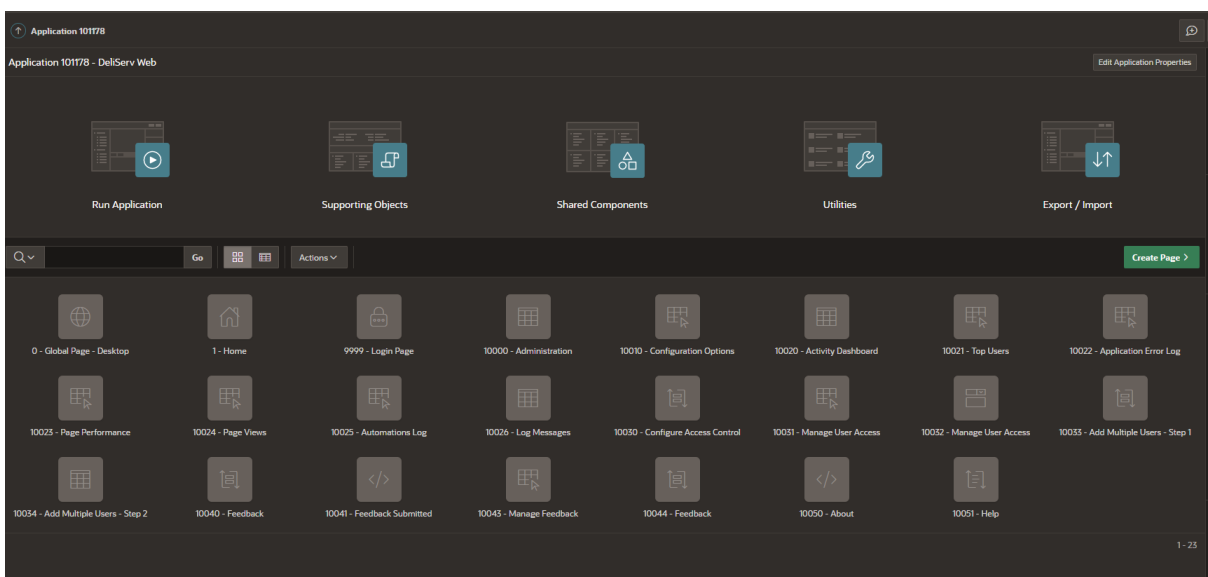
Zatim možemo odabrati *New Application* (kako bi kreirali aplikaciju „od nule“), *uploadati* podatke pa po njima napraviti aplikaciju ili odabrati neki od ponuđenih predložaka aplikacije pa ih urediti po vlastitim potrebama. Biramo *New Application*.

Na sljedećem prozoru, kojeg prikazuje Slika 3, postavljamo temeljne karakteristike nove aplikacije. Biramo ime (DeliServ Web) te izgled korisničkog sučelja (tema i oblik navigacije). Moguće je odmah dodati stranice te omogućiti neke od značajki koje APEX nudi („O aplikaciji“ stranica, stranica za povratne informacije, izvještaji o korištenju i dr.). Dodatno je moguće namjestiti ID aplikacije, naziv sheme, način autorizacije, jezik te dodatne postavke same aplikacije ali i korisničkog sučelja. Odabiremo *Create Application*.



Slika 3. Blueprint aplikacije

Dolazimo do glavnog prozora aplikacije (Slika 4). Ovdje će se prikazati sve stranice koje kasnije kreiramo. Za sada su tu vidljive stranice koje APEX sam generira za slučaj da ih trebamo. Iako na prvu nisu potrebne, dobro ih je ostaviti jer neće utjecati na samu aplikaciju, a mogu biti korisne u kasnijim nadogradnjama.

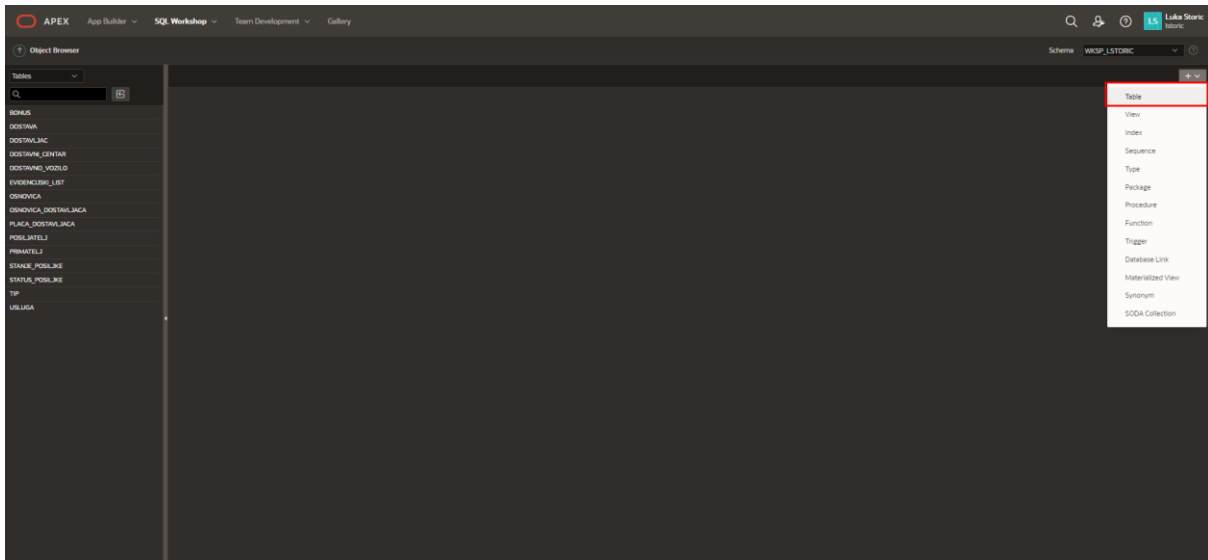


Slika 4. Glavni zaslon

### 4.3. Izrada tablica

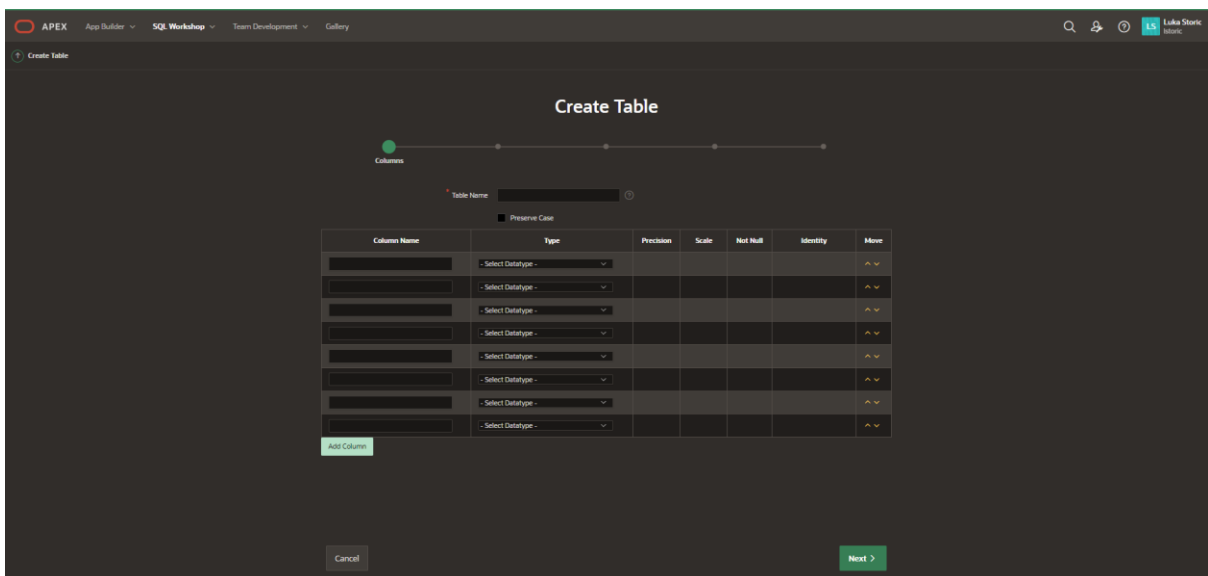
Da bi novokreirana aplikacija mogla raditi te imala smisla, potrebno je definirati bazu podataka. Za to, moramo kreirati tablice. U gornjem dijelu prozora biramo padajući izbornik *SQL Workshop* > gumb *Object Browser*. Ovdje će biti prikazane sve tablice koje kreiramo ali i veliki broj ostalih elemenata (*View, Trigger, Function, Procedure...*).

Za kreiranje nove tablice odabiremo izbornik sa znakom plusa te biramo *Table* (Slika 5).



Slika 5. Object Browser

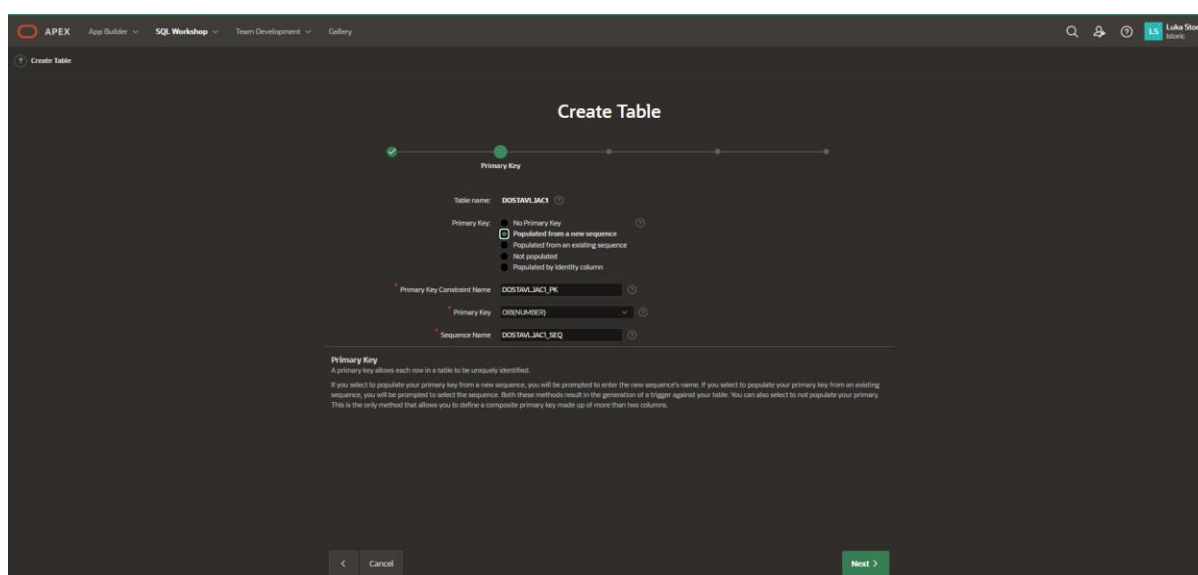
Slika 6 prikazuje sljedeći prozor na kojem odabiremo ime tablice te definiramo stupce. Postavljamo nazive, tip podataka koji će biti spremljeni, veličinu, broj decimalnih mjesta te je li stupac „not null“ tj. može li sadržavati null vrijednost.



Slika 6. Nova tablica

Slijedi postavljanje primarnog ključa. APEX nudi nekoliko opcija:

1. *No Primary Key* – Tablica nema primarni ključ te se nastavlja s postavkama (neupotrebljivo u odabranoj aplikaciji)
2. *Populated from a new sequence* – APEX u pozadini generira sekvencu tj. niz brojeva koji mogu poslužiti za primarni ključ. Korisnik ne mora koristiti taj niz, ali je potreban sustavu za raspoznavanje ključeva. Ovu postavku ćemo koristiti za sve tipove entiteta osim slabog i agregacije. Potrebno je postaviti ime ograničenja (*constraint*), odabrati stupac koji će služiti kao primarni ključ te imenovati sekvencu.
3. *Populated from an existing sequence* – Korištenje postojeće sekvence. Postavke su gotovo iste, ali se ime sekvence bira iz liste postojećih.
4. *Not populated* – U slučaju kada je primarni ključ složen, odabire se ova opcija. Potrebno je odabrati stupce koji sačinjavaju ključ. Ovu opciju koristimo za kreiranje slabog tipa entiteta i agregacije.
5. *Populated by Identity column* – Identity stupac je jedna od novijih funkcionalnosti Oracle baze. Radi se o stupcu kojeg također APEX generira, a koristi se u svrhu auto-inkrementa. U tom stupcu nije dozvoljeno unositi vrijednosti, pa ga je moguće koristiti kao prim-ključ. Postavke su iste, ali je bitno da je u ranijem koraku odabrana Identity postavka na nekom stupcu. Sučelje prikazuje Slika 7.

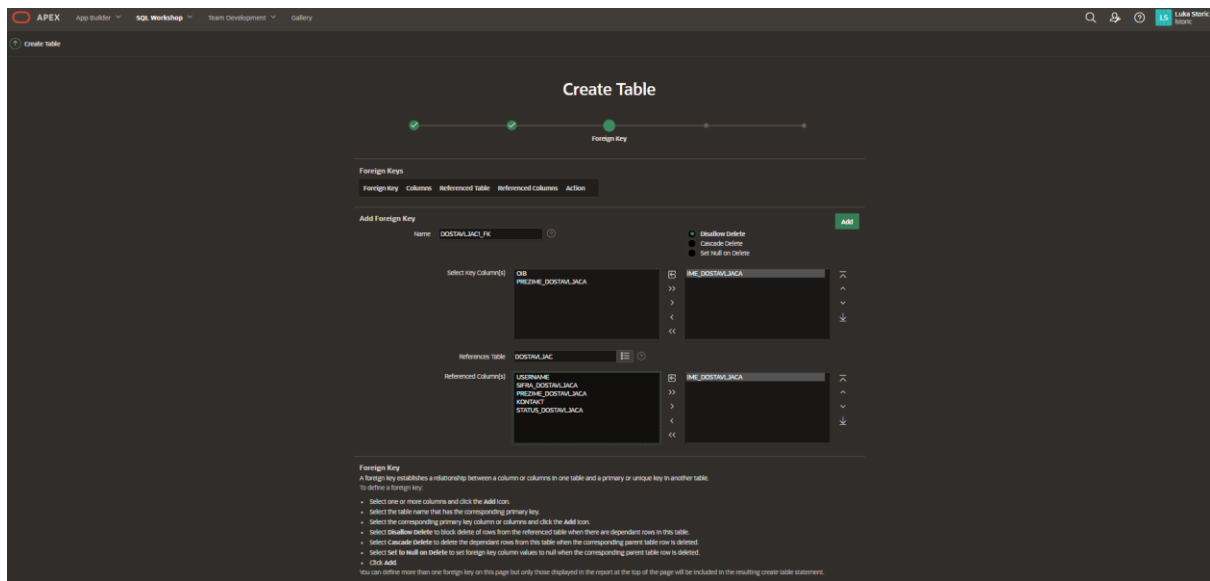


Slika 7. Postavljanje primarnog ključa

Nakon primarnog, potrebno je postaviti vanjske ključeve (ako postoje). Postavljanje, prikazano na Slici 8, dosta je intuitivno; odabire se ime ograničenja tj. samog vanjskog ključa. Zatim se odabire stupac koji ima ulogu vanjskog ključa. U donjem dijelu prozora odabiremo referenciranu tablicu te stupac koji u njoj ima ulogu primarnog ključa. Na kraju je bitno postaviti referencijalni integritet. Radi se o postavki što će se dogoditi ukoliko se redak s primarnim ključem u referenciranoj tablici proba obrisati. Ako odaberemo *Disallow Delete*, brisanje će biti onemogućeno ako u kreiranoj tablici postoji zavisni redak. Odabirom opcije *Cascade Delete*, brisanjem retka u referenciranoj tablici, obrisat će se i zavisni redak u kreiranoj tablici. Treća opcija *Set Null on Delete* umjesto brisanja zavisnog retka, postavlja

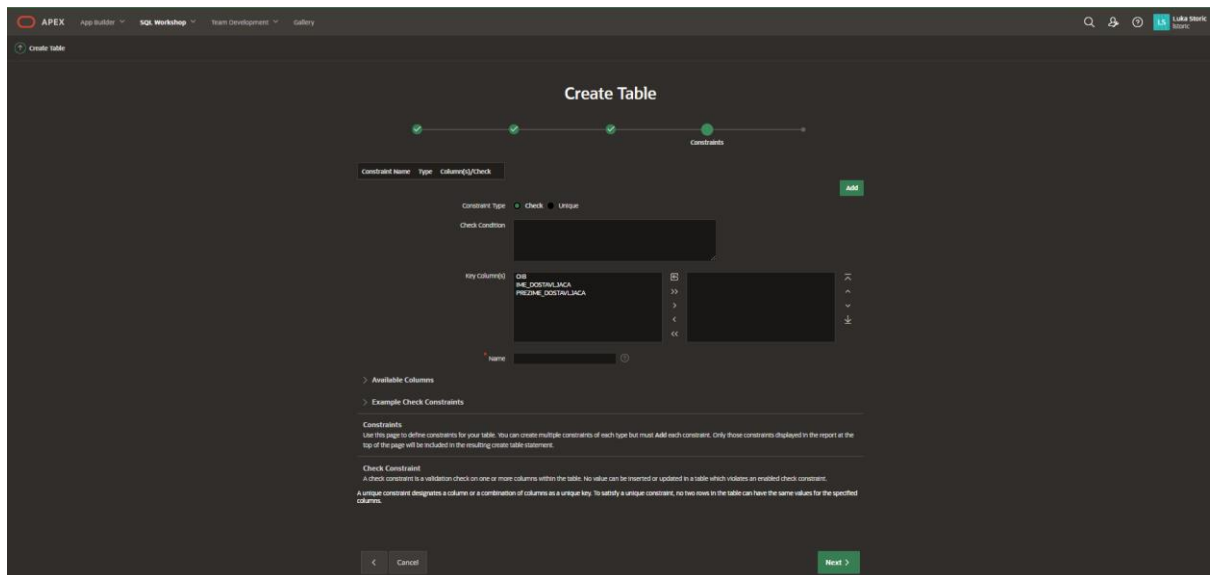


*null* vrijednost u polja koja su bila referencirana. Pritiskom na gumb *Add* dodajemo nove vanjske ključeve, vodeći računa o jedinstvenosti imena.



Slika 8. Postavljanje vanjskog ključa

Na Slici 9 vidimo zadnji korak, a to je dodavanje ograničenja (*Constraints*). Ona služe kao kontrola vrijednosti u određenim stupcima. Ograničenje može biti tipa *Check* (svaka unesena vrijednost se uspoređuje sa zadanim uvjetom te se unos ne dozvoljava ako uvjet nije zadovoljen) ili *Unique* (dva različita retka ne mogu imati istu vrijednost za odabrani stupac/stupce.). Unique ograničenje se automatski generira postavljanjem primarnog ključa. I ovdje je bitno jedinstveno imenovati dodano ograničenje.



Slika 9. Postavljanje ograničenja

Posljednji prozor pokazuje sažetak svih postavki kao i SQL kod kojim će se tablica kreirati. Pritiskom na *Create Table* proces je završen i tablica je kreirana.

Ponavljanjem ovog procesa kreiramo sve potrebne tablice, pazeći pritom na ispravne postavke primarnih ključeva (kod agregacije se primarni ključevi kreiraju i kao vanjski!). Slika 10 pokazuje primjer agregacije. Ograničenja *is not null* (tip *Check*) generirana su nakon označavanja *checkbox*-a „*not null*“ u prvom prozoru kreiranja tablice.

Table	Data	Indexes	Models	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	List Change	Index	Invalid			
OSNOVICA_DOSTAVLJACA_FK	Foreign	-	OSNOVICA_DOSTAVLJACA_FK (WSP_LSTORIC_DOSTAVLJAC)	SIFRA_DOSTAVLJACA	CASCADE	ENABLED	04/17/2021 06:12:27 PM	-	-			
OSNOVICA_DOSTAVLJACA_FK_I	Foreign	-	OSNOVICA_DOSTAVLJACA_FK_I (WSP_LSTORIC.OSNOVICA)	RB_OSNOVICE	CASCADE	ENABLED	04/17/2021 06:12:27 PM	-	-			
OSNOVICA_DOSTAVLJACA_PK	Primary	-	-	SIFRA_DOSTAVLJACA, RB_OSNOVICE	-	ENABLED	04/17/2021 06:12:27 PM	OSNOVICA_DOSTAVLJACA_PK	-			
SVS_COO107167599	Check	"SIFRA_DOSTAVLJACA" IS NOT NULL	-	-	-	ENABLED	04/17/2021 06:12:26 PM	-	-			
SVS_COO107167600	Check	"RB_OSNOVICE" IS NOT NULL	-	-	-	ENABLED	04/17/2021 06:12:26 PM	-	-			

Slika 10. Primjer ključeva kod agregacije

Kreiranje tablica može se obaviti i prije kreiranja aplikacije, ovisno o preferencijama korisnika.

#### 4.4. Izrada korisničkog računa

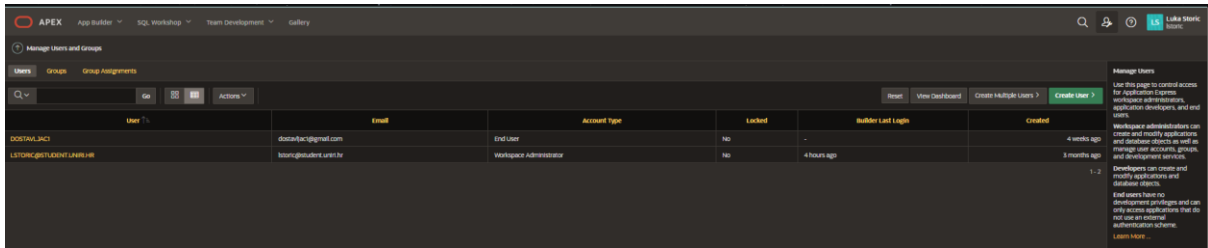
Kako bi aplikaciji mogli uopće pristupiti, kako za testiranje tako i za kasnije korištenje, potrebno je kreirati korisničke račune u samom APEX-u.

Odabiremo padajući izbornik u gornjem desnom kutu *Administration* > gumb *Manage Users and Groups* (Slika 11).

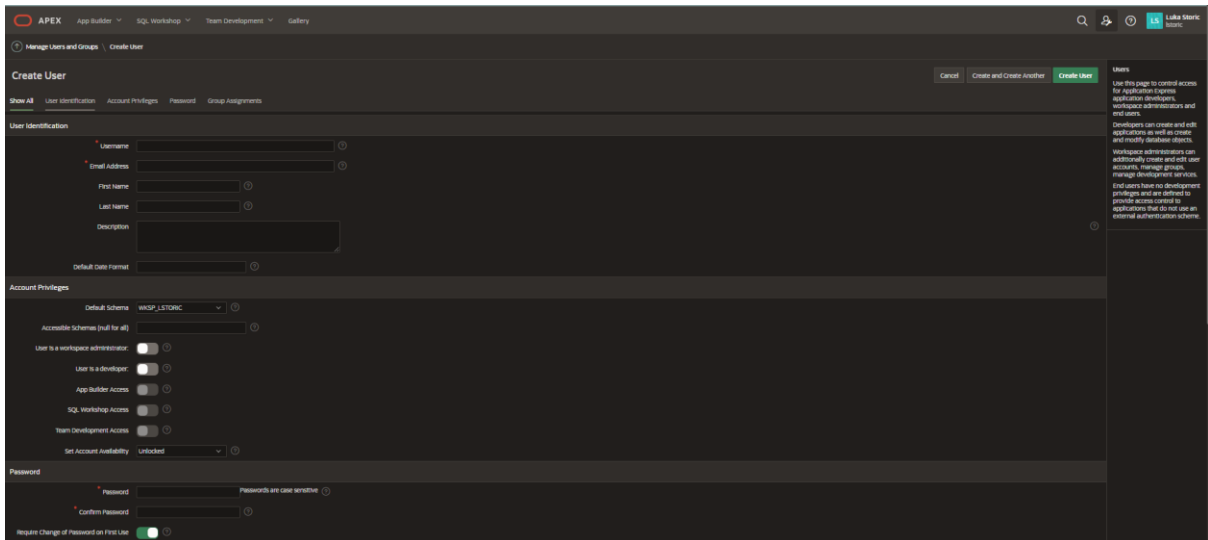
U novom prozoru bit će prikazani svi kreirani korisnički računi. Za dodavanje novog, odabiremo gumb *Create User*.

Slika 12 prikazuje prozor u kojem postavljamo *Username* (korisničko ime) korisnika te email adresu. Dodatno se mogu zapisati ostali osobni podaci (ime, prezime, opis) te je li korisnik administrator ili developer. Ova opcija je bitna zbog pristupa elementima stranice, o čemu će biti riječ u nastavku. Naravno, potrebno je još postaviti lozinku za prijavu te omogućiti opciju obavezne promjene lozinke pri prvoj prijavi. Time je korisnik postavljen.

Kreiramo jedan *admin* račun za lakše razvijanje aplikacije, te jedan koji će predstavljati dostavljača, dakle bez administratorskih prava.



Slika 11. Manage Users and Groups



Slika 12. Kreiranje korisničkog računa

Sada je sve postavljeno i spremno za izradu stranica aplikacije.

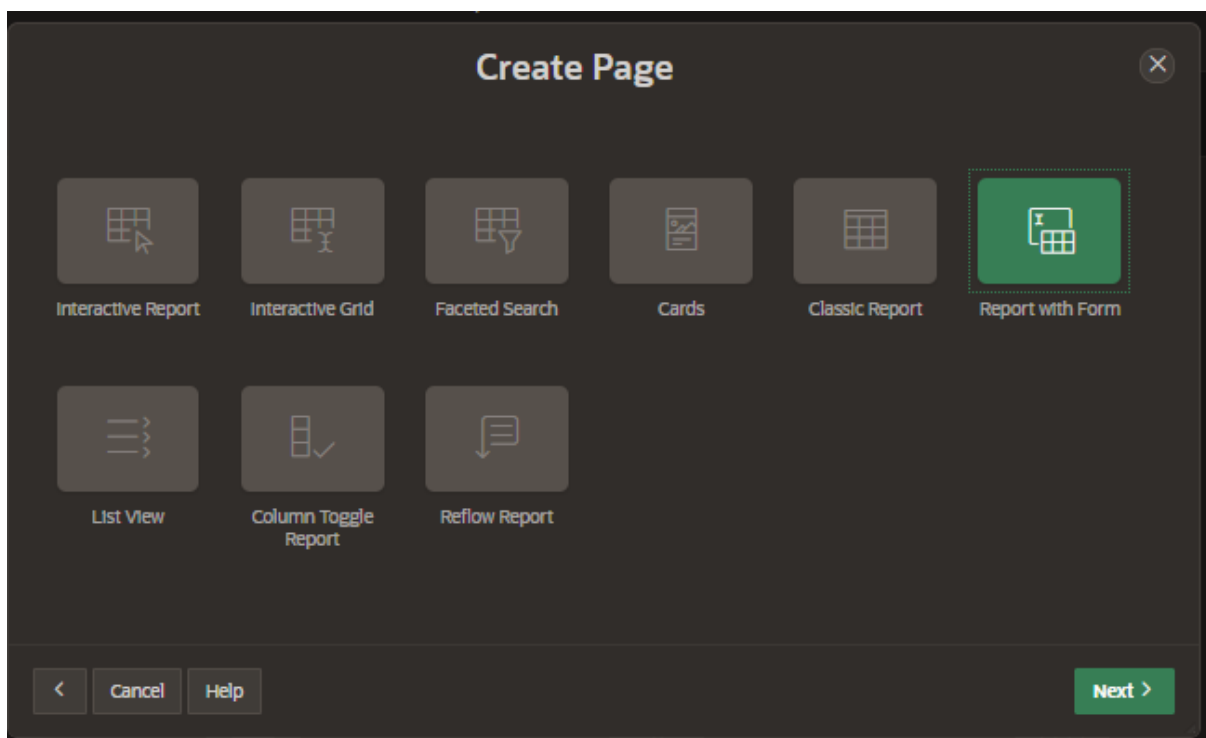
## 4.5. Izrada stranica aplikacije

Proces izrada stranica bit će prikazan u obliku od jednostavnih do složenih stranica. Obzirom da upravo Početna stranica (*Home*) ima najviše sadržaja, ona će biti objašnjena na kraju.

### 4.5.1. Stranice Popis i Unos - kreiranje

Temeljne stranice kod poslovnih aplikacija su upravo oblika popis i unos. Upravo zato, kod kreiranja, APEX nudi istovremenu izradu *Report* i *Form* tipova stranice. U nastavku je objašnjen postupak.

Na glavnoj stranici, biramo tipku *Create Page*. Otvara se novi prozor (Slika 13) u kojem je potrebno odabrati vrstu stranice koju ćemo izraditi. Biramo *Report*. Postoji nekoliko vrsta izvještaja (*report-a*), ali, kao što je ranije rečeno, biramo *Report with Form*.



Slika 13. *Report with Form*

Kao što prikazuje Slika 14, prvi prozor određuje osnovne postavke stranice. Biramo tip izvještaja. *Interactive Report* nudi dodatne mogućnosti koje će biti kasnije naglašene, pa ćemo koristiti taj tip za većinu izvještaja. Odabiremo broj stranice kao identifikator te samo ime stranice. Obzirom da smo odabrali *Report with Form* isto ponavljamo i za *Form* stranicu. Zatim biramo oblik stranice na kojoj će biti *Form* tj. unos. *Normal* oblik je standardni oblik stranice dok *Modal Dialog* otvara vrstu skočnog prozora kako ne bi morali napuštati stranicu s izvještajem. Taj oblik ćemo koristiti za sve stranice Unosa.

Slika 14. *Report with Form* - Osnovne postavke

Sljedeći prozor nudi opcije prikaza u navigaciji. APEX nudi tri opcije:

1. *Do not associate this page with a navigation menu entry* – Za stranice unosa se gotovo nikada ne kreira element navigacije odnosno prečac
2. *Create a new navigation menu entry* – Kreiranje novog elementa navigacije. Pri odabiru ove opcije moramo imenovati element te odabrati nadelement (ako je potrebno) kojem će novi prečac pripadati
3. *Identify an existing navigation menu entry for this page* – U slučaju da element navigacije postoji, ali nije dodijeljen određenoj stranici, možemo dodijeliti novu stranicu tom elementu

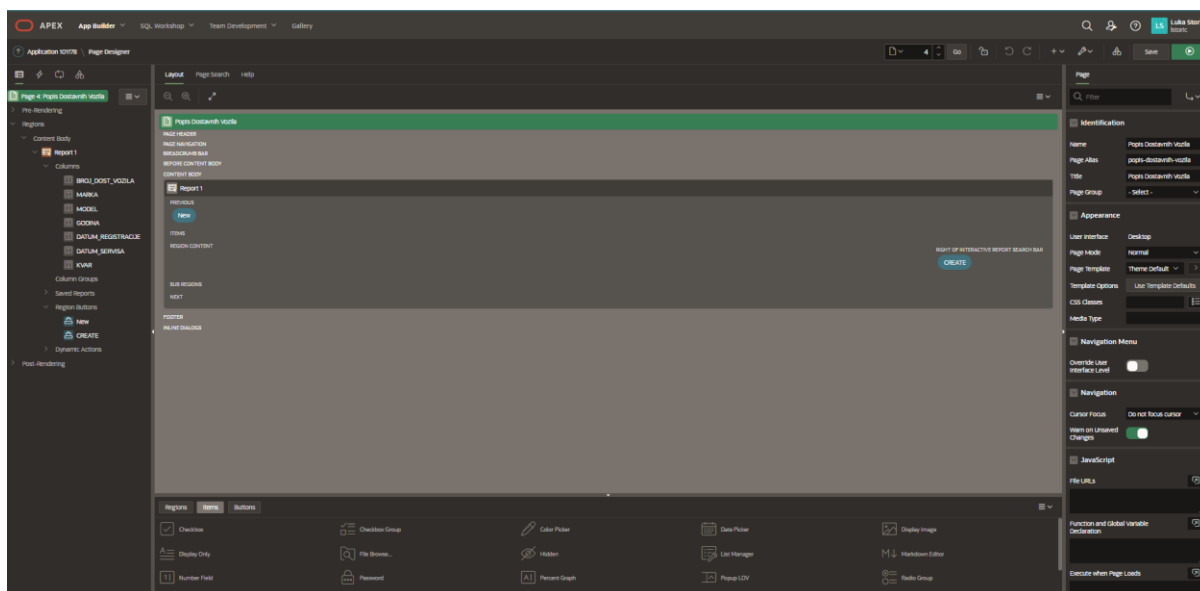
U idućem koraku odabiremo izvor podataka, odnosno tablicu ili više njih nad kojima će se odvijati prikaz i unos podataka. APEX nudi mogućnost *Table* (biramo tablicu te iz nje odabiremo potrebne stupce za izvještaj) ili *SQL Query* (direktno pišemo SQL kod za *select* podataka).

Posljednji prozor je sličnog oblika. Biraju se stupci za koje će se kreirati elementi Obrasca (*Form*). Također, biramo tip primarnog ključa (upravljan od strane baze ili određeni stupac) te odabiremo o kojem se stupcu radi.

Na ovaj način stranica s Izvješćem i pripadna stranica s Obrascem su kreirane. Prelazimo na njihovo uređivanje.

## 4.5.2. Stranice Popis i Unos – uređivanje

APEX-ovo sučelje za uređivanje stranica može se činiti zbudujuće na prvi pogled (Slika 15), međutim prilično je intuitivno. Prati ideju RAD alata gdje se potrebni elementi povlače na centralni dio i pozicioniraju prema željama korisnika. Na slici ispod je *Page Designer* jedne jednostavne stranice s Izvješćem – Popis Dostavnih Vozila.



Slika 15. *Page Designer*

S lijeve strane prozora nalazi se navigacija. Tu su prikazani svi elementi stranice, prema područjima u kojima su pozicionirani. Obzirom da se na ovoj stranici nalazi izvještaj, prikazani su i stupci koje on sadrži. Ovdje također vidimo gumbе korištene na stranici, ali i Dinamičke akcije te Procese o kojima će biti riječ kod složenijih stranica.

U donjem dijelu prozora nalazi se izbornik za regije (*Regions*), predmete (*Items*) i gumbе (*Buttons*). Svaki od njih, prema potrebi, može se povući u stranicu, na željenu poziciju.

Desni dio je najopširniji. U njemu su prikazane sve ponuđene postavke trenutno odabranog elementa stranice ili stranice same po sebi. Ovdje ćemo vršiti najviše promjena.

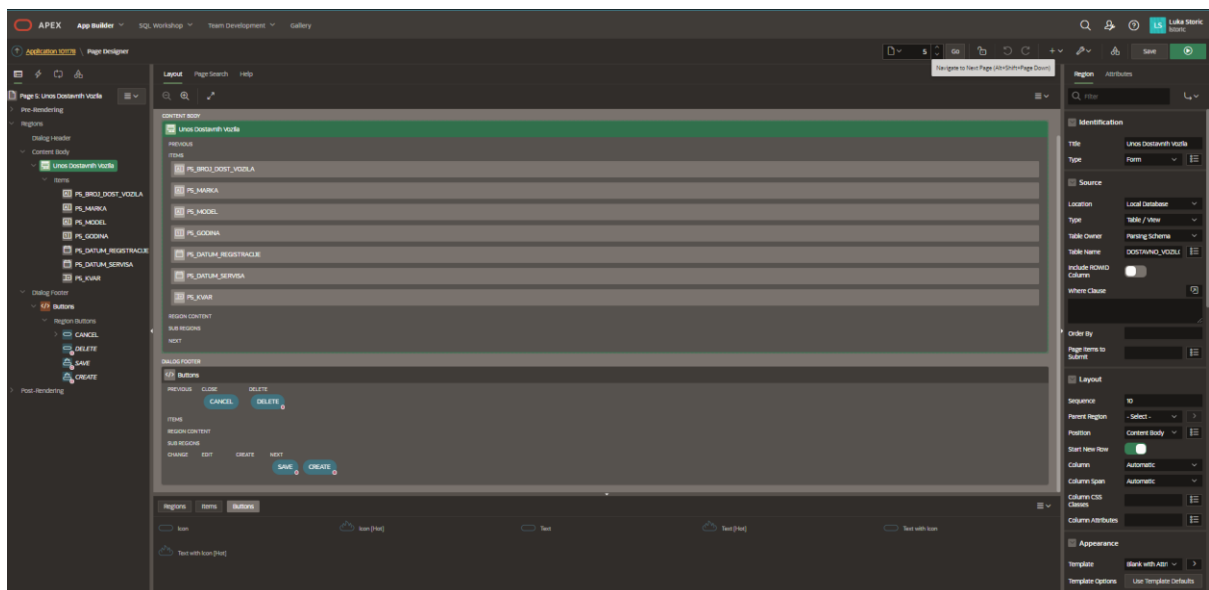
Kod jednostavnih stranica sa izvještajem nisu potrebne velike izmjene nakon kreiranja. Gumbе prevodimo na hrvatski jezik i to tako da ih odaberemo na centralnom dijelu, i onda u postavkama desno promijenimo *Label* u hrvatski izraz. Također na gumbu *Create* u postavkama mijenjamo akciju koju obavlja. Ona je po *defaultu* stavljena na *Submit Page*, a mi ćemo postaviti na *Redirect to Page in this Application*. Pod *Target* stavljamo broj stranice koja identificira novo kreiranu stranicu za unos (*Form*). Dodatno još ubacujemo gumb za povratak na naslovnu stranicu. Iz donjeg izbornika odabiremo gumb *Text [Hot]* i povlačimo na željenu poziciju te također mijenjamo naziv i akciju kako bi gumb preusmjeravao na stranicu 1. Oznaka *Hot* označava da će gumb biti ispunjen bojom koja je odabrana u temi. O promjeni zadane teme biti će govora niže u tekstu.

Stranica Popis Dostavnih Vozila je ovime dovršena. Slika 16 prikazuje izgled u aplikaciji.

	Broj Dostavnog Vozila	Marka	Model	Godina	Datum Istezka Registracije	Datum Servisa	Klasa
	32485	VW	Golf 7	2018	08.08.21	08.08.21	DA

Slika 16. Popis Dostavnih Vozila

Sada je potrebno urediti stranicu za unos. Uzet ćemo isti primjer – Unos Dostavnih Vozila. Sučelje, prikazano na slici 17, slično je, ali vidimo da je APEX generirao regiju *Form* kao i *Static Content* koja sadrži potrebne gumbе (*Cancel*, *Delete*, *Save*, *Create*).



Slika 17. Page Designer - Unos Dostavnih Vozila

Kod stranica s Obrascima možemo primijetiti nekoliko stvari. Nakon što u procesu kreiranja stranice odaberemo primarni ključ, APEX će ga u obrascu automatski sakriti (*Type > Hidden*) kao bi ga zaštitio od izmjena u slučaju da se on automatski generira ili koristi auto-inkrement. Tu postavku možemo vratiti na vrstu podatka koju je potrebno unijeti.

Druga zanimljivost koju sustav sam uređuje su uvjeti kada će se koji gumb prikazati. Ovaj uvjet naziva se *Server-side Condition* i nalazi se pri dnu postavki s desne strane. Kada je taj

uvjet aktivan, element na kojeg se odnosi označen je crvenim kružićem. Kod ovih gumba, uvjet se vodi logikom provjere. Ako *Item P5\_BROJ\_DOST\_VOZILA* nije *null*, uvjet je istinit, te se prikazuje gumb *Save*. U suprotnom prikazuje se samo gumb *Create*. *Server-side Condition* bit će nam potreban za određivanje elemenata stranice koji će biti vidljivi samo administratoru odnosno samo dostavljaču.

Broj dostavnog vozila predviđen je kao peteroznamenasti identifikator. Kako bi osigurali da ta vrijednost ne prelazi 5 znamenki, pri izradi tog stupca odabrana je veličina 5. Međutim sam obrazac neće pokazivati grešku sve dok korisnik ne proba kreirati novo vozilo, a onda će biti potrebno ponovno upisivati sve podatke. Da bi se to izbjeglo, na *Item-u P5\_BROJ\_DOST\_VOZILA* postavljamo *Validation > Maximum Length 5 characters* (Slika 18).

Na taj način, zabranit će se unošenje više od 5 znamenki što će odmah ukazivati korisniku na rečeno ograničenje. Također, u *Inline Help Text* dodajemo objašnjenje polja, kako bi korisnik od početka znao što je potrebno upisati.

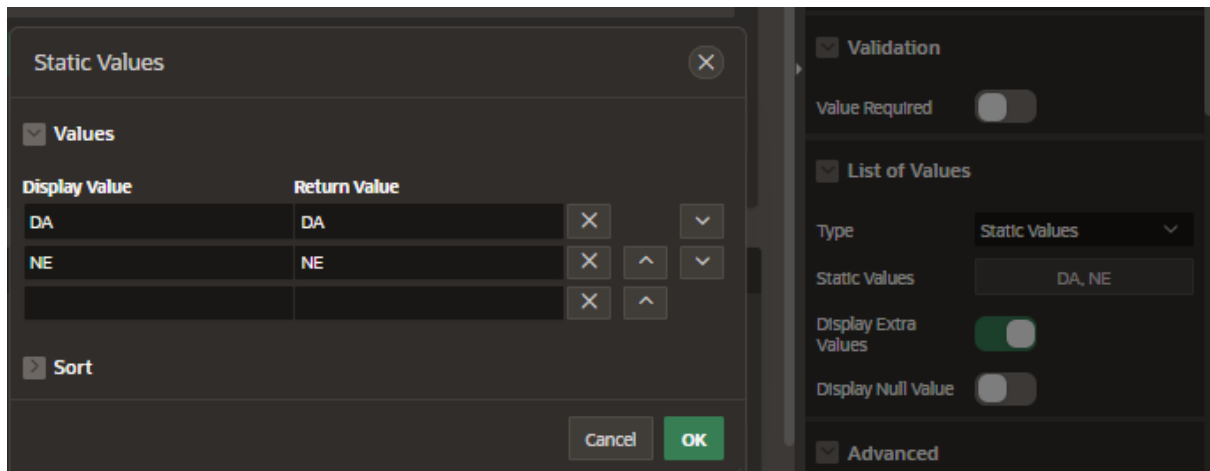


Slika 18. *Validation -Maximum Length*

Pri kreiranju obrasca, sva tekstualna i brojevana polja postavljena su na način da korisnik upisuje vrijednost. To ponekad nije praktično jer želimo dozvoliti samo određene vrijednosti ili vrijednosti iz neke tablice koje korisnik ne mora znati napamet. U tim slučajevima praktično je koristiti *Select List*. Radi se o padajućem izborniku s vrijednostima koje mogu biti *static* tj. unaprijed postavljene ili temeljene na SQL upitu (prikaz vrijednosti iz tablice). U ovom slučaju za polje *P5\_KVAR* koristit ćemo statične vrijednosti 'DA' i 'NE'. Mijenjamo *Type>Select List* te u odjeljku *List of Values* biramo *Type>Static Values*. Ispod toga otvaramo prozor *Static Values*, prikazan na slici 19 te upisujemo vrijednosti DA i NE kako pod *Display Value* tako i pod *Return Value*. Na taj način osiguravamo da ako korisnik odabere DA, ta vrijednost će se upisati i u bazu.

Vrijednosti temeljene na SQL upisu biti će objašnjene kasnije na drugim stranicama.





Slika 19. Select List - Static Values

Stranice Popis Dostavnih Centara, Unos Dostavnog Centra, Popis Statusa Pošiljke, Unos Statusa Pošiljke, Popis Stanja Pošiljke, Unos Stanja Pošiljke, Popis Tipova Pošiljke, Unos Tipa Pošiljke, Popis Dostavljača, Unos Dostavljača, Popis Pošiljatelja, Unos Pošiljatelja, Popis Primatelja, Unos Primatelja, Popis Usluga, Unos Usluga, Popis Osnovica, Unos Osnovica, Popis Bonusa te Unos Bonusa kreirane su na ovaj način. U nastavku će biti opisane preostale, složnije stranice.

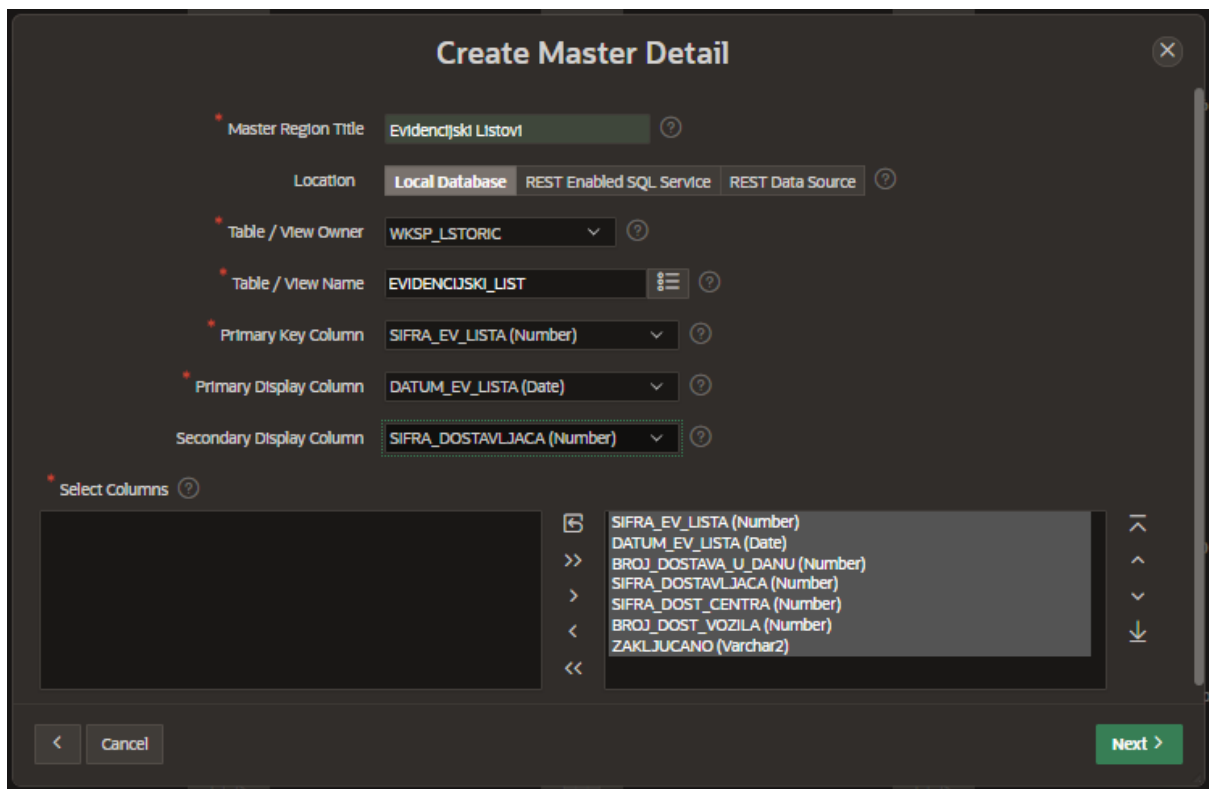
#### 4.5.3. Stranica Popis Evidencijskih Listova (Administrator / Dostavljač)

Stranica Popis Evidencijskih Listova razlikuje se od ostalih prema tipu stranice. Njezina namjena je dati administratoru uvid u sve evidencijske listove te detalje pripadnih dostava. Zato je odabran relativno novi tip stranice: *Master Detail*. Radi se APEX-ovoj inovaciji koja prema postavljenim vezama među dvije tablice, a na osnovu ključeva, prikazuje „glavni podatak“ (*Master*) te povezane podatke (*Detail*).

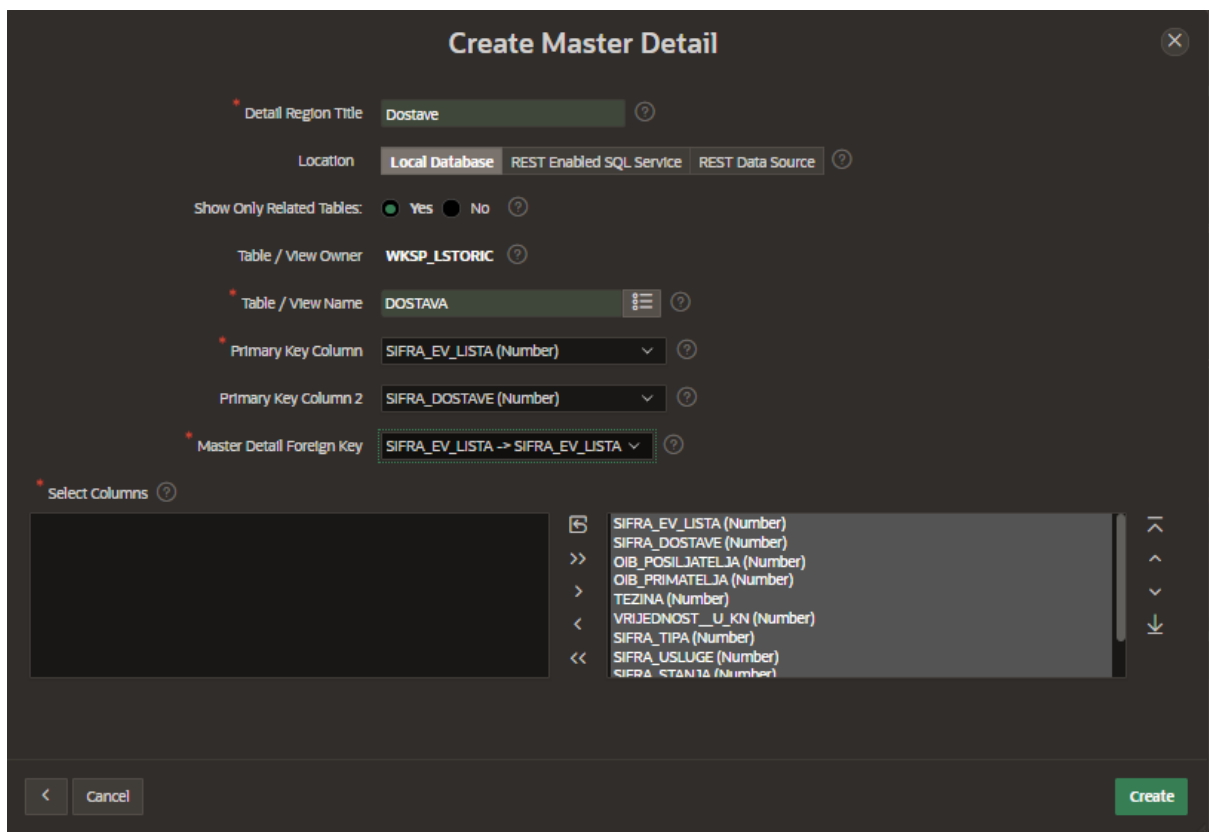
U ovoj aplikaciji, *Master* podaci su zapisi o Evidencijskim Listovima. Osnovni podaci prikazuju se s lijeve strane u obliku liste. Odabirom jednog Lista, s desne strane, prikazuju se detaljni podaci o samom Listu kao i zapisi o svim dostavama tog Lista te oni čine *Detail* podatke. U nastavku je opisan proces izrade te završni izgled stranice.

Da bi kreirali ovakvu stranicu, na glavnoj stranici odabiremo *Create Page* te tip *Master Detail*. Zatim je potrebno odabrati oblik stranice. APEX nudi tri varijante: *Stacked*, *Side by Side* i *Drill Down*. Za potrebe ove aplikacije odabrat ćemo *Side by Side*. U sljedećem prozoru potrebno je odabrati broj stranice (automatski generirano) i naziv. Zadnji prozor služi za odabir izvora stranice. Odabiremo *Master* tablicu te primarni i sekundarni stupac koji će se prikazivati u popisu Listova. U polju *Detail Table* biramo povezanu tablicu sa zapisima o dostavama.

Obzirom da APEX, koristeći svoju logiku, ponekad ne nudi potrebne stupce za prikaz, pritiskom gumba *Advanced* možemo otvoriti napredne postavke. Na taj način, kroz dva nova prozora (Slike 20 i 21), možemo odabrati potrebne stupce za prikaz kako iz *Master* tako i iz *Detail* tablice.



Slika 20. Kreiranje Master Detail stranice - Master tablica



Slika 21. Kreiranje Master Detail stranice - Detail tablica

U izradi je automatski generiran gumb za dodavanje novog zapisa u *Master* tablicu odnosno novog Evidencijskog Lista. Dodatno, dodajemo gumb za ispis. Gumb pokreće Dinamičku akciju temeljenu na JavaScript izrazu, a osigurava da će nepotrebni dijelovi stranice biti skriveni te pokrenut prozor za ispis. Ispod su prikazani spomenuta skripta te, na slici 22, konačni izgled stranice.

```
//// Collapse Left Navigation Bar If It Was Expanded ////
var navCollapsed = 0;
if ($("#body").hasClass("js-navExpanded")) {
    $("#t_Button_navControl").click();
    navCollapsed = 1;
};
setTimeout(3000);

//////// Hide //////////

//Hide Navigation Bar List
$("#t_Header").hide();

//Hide Navigation Menu
$("#t_Body_nav").hide();

//Hide Breadcrumb
$("#t_Body_title").hide();

//Hide Content Offset
$("#t_Body_content_offset").hide();

//Hide Interactive Report Search Bar
$(".a-IRR-toolbar").hide();

//Hide Report Link Column Header
$("#LINK").hide();

//Hide Report Link Column Elements
$('td[headers="LINK"]').hide();

//Hide Buttons
$(".t-Button").hide();

//Hide Footer
$(".t-Footer").hide();

$("#t_Body_side").hide();

$(".a-DevToolbar").hide();

//////// Browser Print //////////
window.print();
```

```

//////// Show //////////

//Show Navigation Bar List
$("#t_Header").show();

//Show Navigation Menu
$("#t_Body_nav").show();

//Show Breadcrumb
$("#t_Body_title").show();

//Show Content Offset
$("#t_Body_content_offset").show();

//Show Interactive Report Search Bar
$(".a-IRR-toolbar").show();

//Show Report Link Column Header
$("#LINK").show();

//Show Report Link Column Elements
$('td[headers="LINK"]').show();

//Show Buttons
$(".t-Button").show();

//Show Footer
$(".t-Footer").show();

//// Expand Left Navigation Bar If It Was Colapsed ////
if ( navCollapsed == 1 ) {
    $("#t_Button_navControl").click();}

```

**Popis Evidencijskih Listova**

09.06.19 2222

09.03.21 2222

21.05.21 2222

03.06.21 2222

05.06.21 3333

09.06.21 2222

Evidencijski Listovi

Detalji Evidencijskog Lista 09.06.19

Broj Dostava U Danu 0

Šifra Dostavnika 22222

Šifra Dostavnog Centra 12345

Broj Dostavnog Vozila 32455

Zaključeno DA

Dostave

Šifra Dostave	OSB Podjela	OSB Primatelj	Većina	Vrijednost U Kn	Šifra Tipa	Šifra Usluge	Šifra Stanja	Šifra Statusa	Cijena Dostave Ukupno
12	T21*19803	T21*19803			1	22222	2	1	

Slika 22. Popis Evidencijskih listova

Kao što je rečeno ranije, ova stranica služi administratoru za uvid u sve Evidencijske listove. Kreirana je i stranica sličnog naziva (Popis Evidencijskih listova – Dostavljač) koja omogućuje dostavljaču da pregleda podatke samo o svojim proteklim Listovima.

Radi se o jednostavnoj stranici koja sadrži jedan izvještaj. Sam izvještaj definiran je WHERE uvjetom kako bi prikazivao podatke o Evidencijskim Listovima isključivo prijavljenog dostavljača. Slika 23 prikazuje postavku *Where Clause*.

```
(sifra_dostavljacka in (select sifra_dostavljacka from dostavljack, dual  
where Upper(dostavljack.username)=:APP_USER))
```



Slika 23. Popis Evidencijskih Listova – Dostavljač

Na isti način kreirana je i stranica Popis Dostava, ali bez mogućnosti ispisa.

#### 4.5.4. Stranica Unos Evidencijskog Lista

Unos Evidencijskog Lista također je jednostavna stranica, ali je vrijedno spomenuti polja obrasca koja su definirana kao *Select List* tip. U njima je korišten SQL kod za dobivanje vrijednosti u padajućem izborniku. Radi se o poljima P33\_SIFRA\_DOST\_CENTRA i P33\_BROJ\_DOST\_VOZILA.

Za dobivanje šifre dostavnog centra, korišten je sljedeći kod:

```
select sifra_dost_centra || ' ' || naziv_dost_centra d, sifra_dost_centra v  
from dostavni_centar;
```

Oznake *d* i *v* služe kako bi APEX prepoznao koja vrijednost je prikazana (*display*), a koja će biti zapisana u tablicu (*value*).

#### 4.5.5. Stranica Unos Dostava

Iako se radi o jednostavnoj stranici s obrascem, razlika je u tome što se povlače podaci iz raznih tablica pa je tek u nekoliko polja dozvoljen slobodan unos.

Za podatke koji su zapisani u drugim tablicama, koristi se *Select List* kako bi se osigurao integritet. Jedan od ključeva tablice Dostava, Sifra\_ev\_lista, dobiva se iz tablice Evidencijski List kroz sljedeći SQL kod:

```
select sifra_ev_lista d, sifra_ev_lista v from evidencijski_list where sifra_dostavljacka in (select sifra_dostavljacka from dostavljack,dual where Upper(dostavljack.username)=:APP_USER) and zakljucano='NE' ;
```

Kombinacijom uvjeta gdje *Username* mora biti trenutno logirani korisnik te zastavica Zakljucano postavljena na NE, osigurano je da se automatski ispuni ispravna šifra Evidencijskog Lista.

Kako bi osigurali da korisnik neće promijeniti tu vrijednost, polje P26\_SIFRA\_EV\_LISTA postavljena je na tip *Read Only* (Slika 24)



Slika 24.Unos Dostave

#### 4.5.6. Stranica Popis Osnovica Dostavljača

Obzirom da je relacija OSNOVICA\_DOSTAVLJACA agregacija, sadrži samo ključeve povezanih tablica. Zato je kroz SQL kod postignuto da izvještaj sadrži više podataka kroz veći broj prikazanih stupaca. Konačni izgled prikazuje slika 25.

```
select sifra_dostavljacka, (select ime_dostavljacka from dostavljack d where o.sifra_dostavljacka=d.sifra_dostavljacka) as ime_dostavljacka,
```

```
(select prezime_dostavljacka from dostavljack d where o.sifra_dostavljack
a = d.sifra_dostavljacka) as prezime_dostavljacka,
rb_osnovice,
(select iznos_osnovice from osnovica os where o.rb_osnovice =
os.rb_osnovice) as iznos_osnovice
from osnovica_dostavljacka o;
```

Ime Dostavljacka	Prezime Dostavljacka	Iznos Osnovice
Marko	Marko	\$2,000.00
Luka	Luka	\$2,000.00
Pero	Pero	\$2,000.00
Ivo	Ivo	\$2,000.00

Slika 25. Osnovice Dostavljača

#### 4.5.7. Stranica Unos Osnovice Dostavljača

Obrazac na ovoj stranici je jednostavan, kroz *Select List* biraju se šifra dostavljača te redni broj osnovice. Ipak, kako bi se izbjegli duplikati ili situacije gdje je istom dostavljaču dodijeljeno više osnovica, dodan je WHERE uvjet u SQL kod koji osigurava da ponuđene šifre ne postoje već u tablici Osnovica\_dostavljacka.

```
select ime_dostavljacka || ' ' || prezime_dostavljacka d, sifra_dostavljacka v
from dostavljack where sifra_dostavljacka not in (select sifra_dostavljacka from
osnovica_dostavljacka);
```

#### 4.5.8. Stranica Popis Plaća Dostavljača

U izvještaju s ove stranice nailazimo na istu problematiku kao i kod osnovica dostavljača tj. agregacija sadrži samo ključeve povezanih tablica te jedan dodatni stupac za ukupan iznos plaće. Kako bi pregled imao smisla, potrebno je dodati stupce iz ostalih tablica. Najefikasniji način je putem SQL koda.

```
select a.ime_dostavljacka,
a.prezime_dostavljacka,
b.iznos_osnovice,
c.iznos_bonusa,
```

```

d.placa_ukupno
from
  dostavljac a,
  osnovica b,
  bonus c,
  placa_dostavljacka d
where
  a.sifra_dostavljacka=d.sifra_dostavljacka and
  b.rb_osnovice=d.rb_osnovice and
  c.rb_bonusa=d.rb_bonusa;

```

#### 4.5.9. Stranica Unos Plaće Dostavljača

Za ovu stranicu se može reći da je najsloženija ne toliko zbog elemenata (kojih ima tek nekoliko) već zbog pozadinskih procesa koje izvršava pri unosu podataka. Da bi se dobio iznos ukupne plaće određenog dostavljača potrebne su 4 Dinamičke akcije te 2 pomoćna polja.

U nastavku je detaljni opis procesa izračuna i upisa plaće.

1. Korisnik iz *Select List* polja odabire dostavljača
2. Dinamička akcija *Dynac* povratnu vrijednost (šifru dostavljača) postavlja u skriveno polje SFRDOST
3. Korisnik klikom na polje P30\_RB\_OSNOVICE aktivira Dinamičku akciju *Dynac\_1* koja na temelju polja SFRDOST pronalazi redni broj osnovice te postavlja vrijednost u kliknuto polje
4. Korisnik iz sljedećeg *Select List* polja odabire redni broj bonusa
5. Dinamička akcija *Dynac\_3* povratnu vrijednost (redni broj bonusa) postavlja u skriveno polje RBBON
6. Korisnik klikom na polje P30\_PLACA\_UKUPNO aktivira Dinamičku akciju *Dynac\_2* koja na temelju SQL koda (ispod) izračunava plaću te postavlja vrijednost u kliknuto polje

```

select (a.iznos_osnovice + b.iznos_bonusa) from osnovica a, bonus b
where a.rb_osnovice =:P30_RB_OSNOVICE and b.rb_bonusa =:RBBON;

```

Sam proces izrade Dinamičkih akcija relativno je jednostavan. U izborniku Dinamičkih akcija odabiremo događaj (*Event*) nad kojim se pokreće akcija te kliknemo desnom tipkom miša i biramo *Create Dynamic Action*. U postavkama, u polju *When* biramo element stranice koji će služiti za aktivaciju. Zatim je potrebno ponovno u izborniku Dinamičkih akcija odabrati akciju koja će se dogoditi u slučaju TRUE (zapravo obavezno nakon definiranog događaja) te FALSE ako slučaj postoji (događaj nije postignut). Omogućen je veliki broj



akcija među kojima su *Clear, Disable, Enable, Submit, Refresh, Set Value* itd. Popis Dinamičkih akcija prikazan je na slici 26.

Za dobivanje povratne vrijednosti iz *Select List* polja koja se postavlja u skriveno polje koristila se akcija na temelju JavaScript izraza.

```
apex.item('P30_SIFRA_DOSTAVLJACA').getValue()
```



Slika 26. Unos Plaće Dostavljača - Dinamičke akcije

#### 4.5.10. Stranica Metrike

Stranica Metrike razlikuje se od ostalih po tome što ne sadrži ni izvještaje ni obrasce već tri grafikona (*Chart*). Stranici se pristupa preko Početne stranice za administratore o kojoj će biti riječ u sljedećem poglavlju.

Stranica, prikazana na slici 27, oblikovana je na način da kroz spomenuta tri grafikona prikazuje broj dostava po mjesecima u godini, broj Evidencijskih Listova po mjesecima u godini (stupčasti dijagrami) te dostavljače po broju dostava za protekli mjesec (tortni grafikon).

Obzirom da korištenjem *Select List* polja, APEX vrijednost godine dobiva tek pri slanju stranice, dolazimo do pojave gdje se prikazuju podaci za godinu odabranu prije trenutno odabrane. Kako bi to izbjegli, napravljeno je upisno polje gdje korisnik unosi godinu, potvrđuje tipkom Enter te se podaci prikažu. Na ovaj način prikazuju se podaci u oba stupčasta dijagrama.

Stranica je kreirana na način da smo na glavnoj stranici odabrali *Create Page > Blank Page*. Odaberemo ime stranice i oblik *Modal Dialog* te stranicu ne dodjeljujemo navigaciji. U samom *Page Designer*-u iz donjeg izbornika biramo regiju *Chart*. U izborniku s lijeve strane prikazuje nam se novi sloj (*Series*) kojeg odaberemo te u postavkama s desne strane, u rubrici *Source* biramo *Type > SQL Query* i unosimo sljedeći kod:

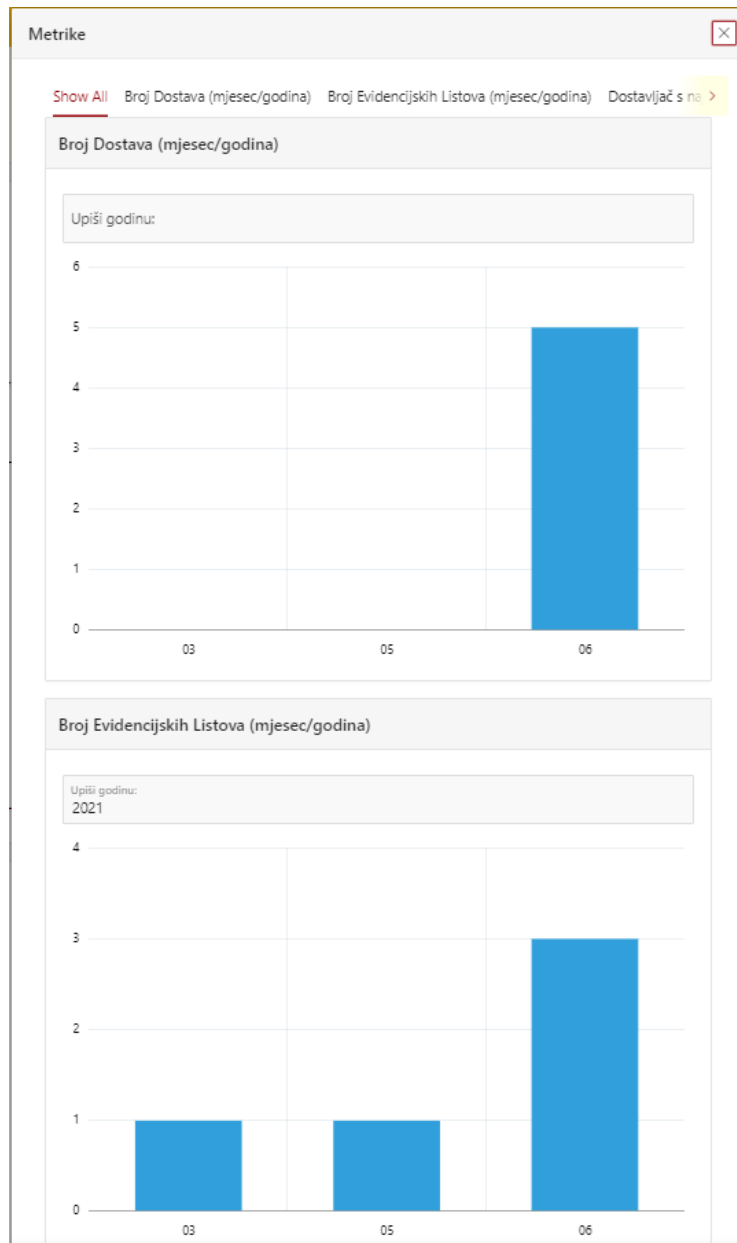
```
select to_char(datum_ev_lista,'MM'), sum(broj_dostava_u_danu)
from evidencijski_list where to_char(datum_ev_lista,'YYYY') like :P28_NEW
group by to_char(datum_ev_lista,'MM');
```

P28\_NEW je gore spomenuto polje za unos godine. Na isti način kreiramo i dijagram za Evidencijske Listove gdje SQL kod glasi:

```
select to_char(datum_ev_lista,'MM'), count(*)
from evidencijski_list where to_char(datum_ev_lista,'YYYY') like :P28_NEW_2
group by to_char(datum_ev_lista,'MM');
```

Tortni dijagram sa podacima o dostavljaču s najviše dostava koristi kod:

```
select sifra_dostavljacka, count(*)
from evidencijski_list
where trunc(datum_ev_lista,'MM') = add_months(trunc(sysdate,'MM'),-1)
group by sifra_dostavljacka;
```



Slika 27.Metrike

#### 4.5.11. Početna stranica (Administrator / Dostavljač)

Kao što je rečeno na početku, Početna stranica sadrži najviše elemenata. Iako se radi o jednoj stranici, *Server-side Condition* uvjetima određeno je koje elemente vidi administrator, a koje dostavljač što daje privid dvije različite stranice. Pri kreiranju aplikacije, APEX generira Početnu stranicu bez sadržaja, jedino je vidljiv naziv aplikacije i logo. Krenimo od sadržaja vidljivih administratoru.

U *Page Designer*-u, u *Regionu* koji sadrži naziv aplikacije, dodajemo *Item* vrste *Display Only* koji će služiti kao pozdravna poruka. U postavkama, u odjeljku *Source*, postavljamo *Type > Static Value* te upišemo pozdravnu poruku za administratora.

Sljedeće, iz donjeg izbornika, dodajemo novi *Region* te ga imenujemo Evidencijski listovi. Pod *Appearance* postavljamo *Template Options* na *Region Title > Level 1*. Ovu opciju koristit ćemo u svim regijama kako bi naslov bio napisan velikim slovima. Prvo kreiramo gumb koji će voditi na ranije opisanu stranicu Metrike. Zatim dodajemo 4 nova *Item*-a koji će prikazivati dinamički generirane podatke na temelju SQL koda. Postavka se nalazi u odjeljku *Source, Type > SQL Query*. Prvi će prikazati broj Evidencijskih Listova s današnjim datumom.

```
select count(*) from evidencijski_list where datum_ev_lista like sysdate;
```

Sljedeći prikazuje ukupan broj dostava s današnjim datumom.

```
select count (*) from dostava where sifra_ev_lista in(select sifra_ev_lista from evidencijski_list where datum_ev_lista like sysdate);
```

Treći prikazuje broj dostava s neoriginalnim stanjem.

```
select count(*) from dostava where sifra_ev_lista in (select sifra_ev_lista from evidencijski_list where datum_ev_lista like sysdate) and sifra_stanja != 1;
```

Posljednji prikazuje broj nedostavljenih dostava.

```
select count(*) from dostava where sifra_ev_lista in (select sifra_ev_lista from evidencijski_list where datum_ev_lista like sysdate) and sifra_statusa != 1;
```

Ispod zadnjeg i predzadnjeg, u slučaju da je broj različit od nule, prikazuje se izvještaj s podacima o dostavi. Da bi to osigurali, u postavkama pod *Server-side Condition*, odabiremo *Type > Item!=Value*, *Item > PI\_NEW\_8*, *Value > 0*.

Sljedeća regija prikazuje broj nezaključanih Evidencijskih Listova te izvještaj s podacima o takvim Listovima. Regija je također definirana *Server-side Condition*-om (*Type > Rows returned*) za *query*:

```
select * from evidencijski_list where zakljucano='NE' and datum_ev_lista<sysdate-1;
```

Nakon toga, kreiramo regiju imena Dostavljači. Postavljamo *Item* koji na prikazuje ukupan broj dostavljača te još jedan koji upozorava na razliku između broja dostavljača u tablici Dostavljač te broja dostavljača po Centrima zapsanog u tablici Dostavni Centar. Dodajemo i izvještaj koji prikazuje neaktivne dostavljače te 2 *Item*-a koji upozoravaju na 30% odnosno 50% neaktivnih dostavljača. Oba *Item*-a pojavljuju se na osnovu *Server-side Condition*-a koji provjeravaju postotak dostavljača sa statusom različitim od „A“. Ispod toga dodajemo i graf koji prikazuje broj dostavljača po statusima.

Iduća regija zove se Dostavni Centri. Dodan je *Item* koji broji Dostavne Centre te dijagram koji prikazuje broj dostavljača i broj vozila po Centrima. U dijagramu kreiramo dva *Series* sloja s postavkama u odjeljku *Source: Type > SQL Query*

```
select naziv_dost_centra label, broj_dostavljac from dostavni_centar;
```

te u odjeljku *Column Mapping: Label > LABEL*, *Value > BROJ\_DOSTAVLJACA*

U drugom sloju se koristi SQL kod

```
select naziv_dost_centra label, broj_vozila from dostavni_centar;
```

i postavke *Column Mapping*: *Label* > *LABEL*, *Value* > *BROJ\_VOZILA*

Posljednja regija je Dostavna Vozila. Jedan *Item* broji sva Vozila, sljedeći upozorava na razliku između broja Vozila u tablici Dostavna Vozila te broja zapisanog po Centrima u tablici Dostavni Centar. Ispod toga kreiramo izvještaj koji prikazuje vozila u kvaru s *Item*-om koji upozorava da postoje vozila u kvaru. Kao i do sada, prikaz ovog izvještaja kontrolira *Server-side Condition*. Isto vrijedi i za iduća dva izvještaja. Jedan prikazuje Vozila kojima će uskoro isteći registracija (na osnovu SQL kod provjere u postavci *Source: Where Clause* >

```
(datum_registracije between sysdate and sysdate+7).
```

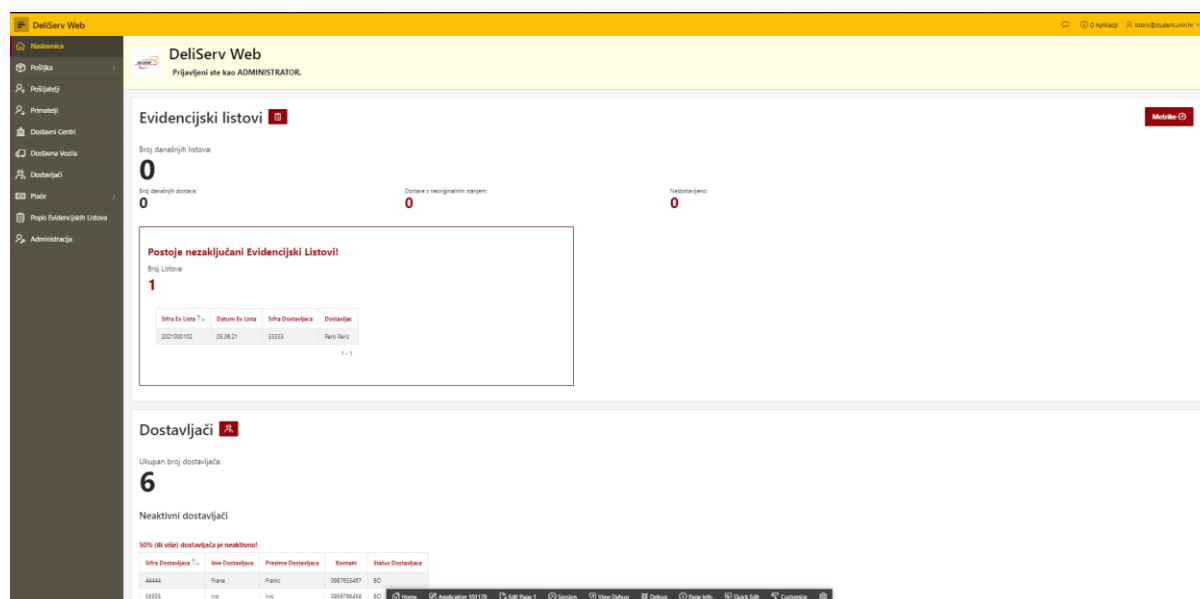
Drugi prikazuje Vozila kojima je istekla registracija na osnovu provjere gdje je

*Where Clause* >

```
(sysdate > datum_registracije)
```

Sve ovo čini Naslovnicu za administratora. Sve regije imaju postavku *Security: Authorization Scheme* > *Administration Rights* kako bi osigurali da upravo jedino administrator vidi navedene regije.

Konačni izgled Početne stranice prikazuje slika 28.



Slika 28. Početna stranica – Administrator

U nastavku slijedi opis regija Početne stranice koje vidi dostavljač.

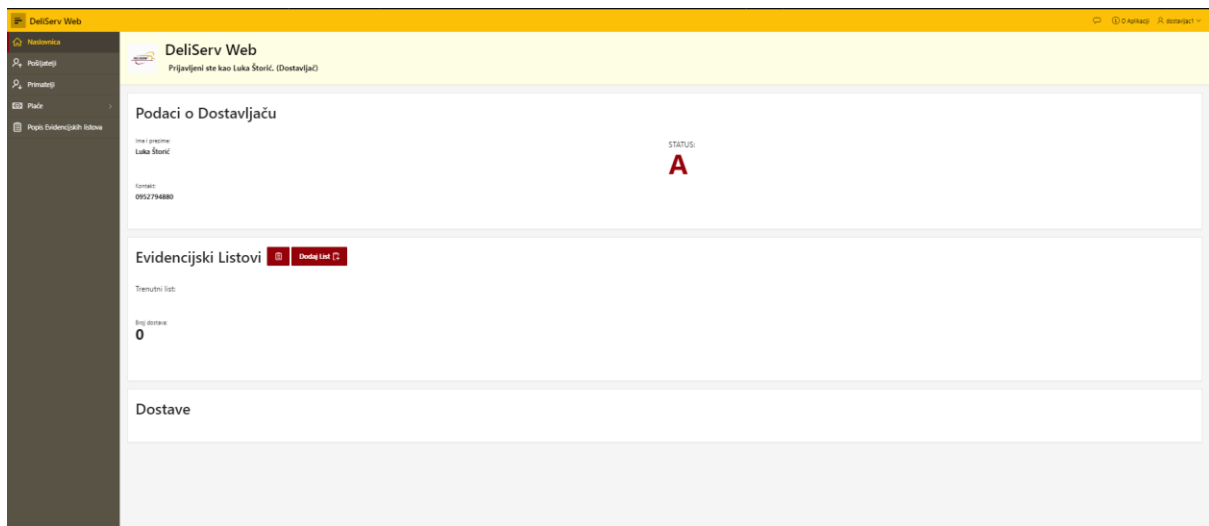
Prva takva regija zove se Podaci o Dostavljaču. Kako joj i ime govori, ona prikazuje osnovne podatke dostavljača točnije ime, prezime, kontakt i status. Za to kreiramo tri *Item*-a definirana SQL kodom koji uspoređuju korisničko ime trenutno prijavljenog dostavljača s tablicom Dostavljač.

```
select ime_dostavljacka || ' ' || prezime_dostavljacka from dostavljack  
where Upper(username)=:APP_USER;
```

Sljedeću regiju imenujemo Evidencijski Listovi. Ovdje je korištena kombinacija *Server-side Condition*-a kako bi se *Item*-i i određeni gumbi prikazivali ovisno o postojanju/zaključanosti Lista. Ukoliko u bazi ne postoji List dodijeljen prijavljenom dostavljaču sa zastavicom Zaključano postavljeno na NE, prikazuje se gumb Dodaj List. Zatim kreiramo dva *Item*-a, jedan koji prikazuje šifru aktivnog Lista te drugi s brojem dostava na aktivnom Listu. Dodajemo gumb ZAKLJUČAJ LIST koji se prikazuje ako je zadovoljen uvjet da postoji List dodijeljen prijavljenom dostavljaču sa zastavicom Zaključano postavljeno na NE. Konačno dodajemo i *Item* koji upozorava dostavljača da je dužan zaključati listu te da nakon zaključavanja izmjene nisu dozvoljene.

Posljednja regija ovdje zove se Dostave. Ona se sastoji samo od gumba za dodavanje dostave, a također je definiran uvjetom da postoji aktivni Evidencijski List.

Finalni izgled prikazan je na slici 29.



Slika 29. Početna stranica – Dostavljač

Sve regije predviđene za dostavljača imaju postavku *Security: Authorization Scheme* > *{Not Administration Rights}* kako bi se osiguralo da se prikazuju jedino dostavljačima.

Bitno je naglasiti da sve regije, kako vidljive administratoru tako i one vidljive dostavljaču, imaju gumb koji vodi na stranicu sa pregledom pripadnih podataka.

#### 4.5.12. Stranice O Aplikaciji i *Feedback*

Stranica O Aplikaciji jednostavna je stranica sa statičnim elementima u kojima je opisan zadatak i sama aplikacija. Link na stranicu nalazi se u alatnoj traci aplikacije.

Stranica *Feedback* (Slika 30) automatski je generirana od strane APEX-a te daje mogućnost korisniku da pošalje povratne informacije i prijedloge developerima stranice.

Povratne informacije

Iskustvo

Povratne informacije o Aplikaciji

Odustani Pošalji Mišljenje

Slika 30.Feedback

#### 4.5.13. Stranica Login

Konačno, stranica Login također je automatski generirana, a služi za prijavu u sustav. Slika 31 prikazuje izgled stranice.



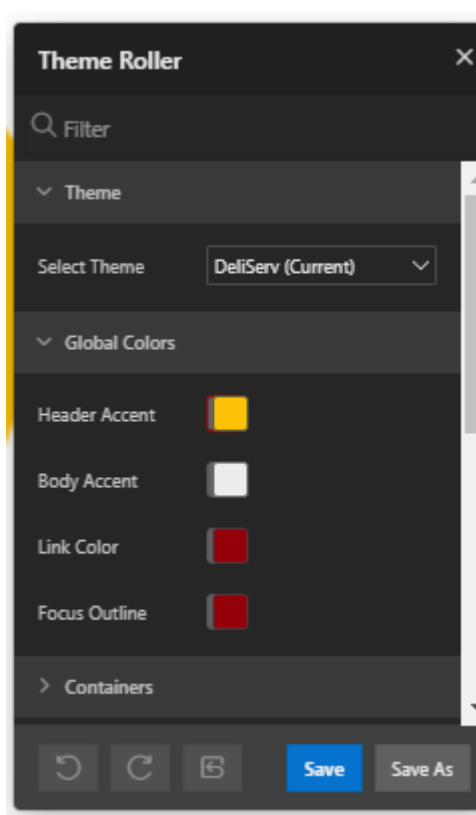
Slika 31. Stranica Login

## 4.6. Grafički dizajn stranica

Za postavljanje grafičkog dizajna APEX ima elegantno rješenje. U pokrenutoj aplikaciji, cijelo vrijeme je u dnu prikazan tzv. *Dev Toolbar* koji sadrži korisne prečace za developera. Među njima nalazi se i gumb *Customize > Theme Roler*.

*Theme Roler* (Slika 32) pruža mogućnost izmjene trenutne teme ili kreiranje nove. Pri tome, podijeljeno po elementima aplikacije, omogućuje odabir glavnih i sporednih boja heksadekadskim zapisom ili odabirom iz palete. Na taj način osiguravamo da će odabrani spektar boja biti distribuiran na cijelu aplikaciju.

Također, u svim elementima stranica, omogućeno je postavljanje CSS klasa ili tzv. *Custom Attributes* gdje se CSS promjene mogu upisati *inline*. Ovakvo uređivanje korišteno je na nekoliko mjesta kroz aplikaciju (Stupci Ukupno, zastavice, Status\_dostavljača itd.)



Slika 32. *Theme Roler*

Ovime su opisani svi aspekti DeliServ Web aplikacije. U nastavku će biti ukratko opisani alati Oracle APEX te SoftVelocity Clarion.



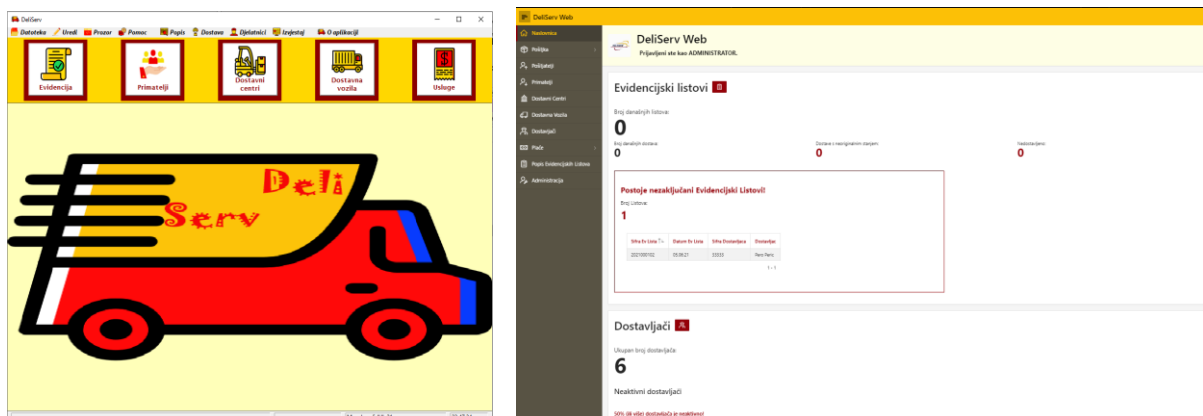
## 5. Usporedba alata Oracle APEX i SoftVelocity Clarion

Usporediti ova dva alata nije jednostavno. Obzirom da su nastali u razmaku od 18 godina prirodno je da se i razlikuju u velikom broju karakteristika i mogućnosti. Ipak, glavna ideja ostaje ista: Kao RAD alat, omogućiti kreiranje aplikacija nad bazom podataka na što jednostavniji i brži način.

### 5.1. SoftVelocity Clarion

Clarion tvrtke SoftVelocity nastao je 1986. godine zajedno s istoimenim jezikom kojeg koristi za prevođenje uputa od strane korisnika u izvorni kod. Kroz 35 godina razvilo se 12 verzija programa, svaka uvodeći neku novu funkcionalnost. Ipak, glavna namjena ostala je razvijanje *desktop* aplikacija nad relacijskim bazama podataka.

Sam softver notorno je poznat po svojoj nestabilnosti te neobjašnjivim greškama. Opasnost od gubitka rada je konstantna te je potrebno učestalo spremanje kreiranog. Također, grafički dizajn aplikacija sveden je na minimum. Korištenje „naprednih“ mogućnosti pruža za današnje standarde zastarjele rezultate. Slika 33 prikazuje usporedbu izgleda Početne stranice DeliServ *desktop* aplikacije kreirane u Clarionu sa Početnom stranicom kreiranom u APEX-u.



Slika 33. Grafički dizajn - Clarion i APEX

Ono što je također bitno naglasiti kod Clariona su licence. Niti za jednu inačicu softvera ne postoje probne verzije, odnosno potrebno je platiti čak i za najjednostavnije vrste aplikacija. Kod privatne upotrebe to može predstavljati problem, naročito kod studenata koji moraju softver nabaviti iz drugih (nelegalnih) izvora.

Ipak, ne može se reći da Clarion sam po sebi ne izvršava svoju zadaću. Kod samog kreiranja aplikacije, pokazao se kao koristan RAD alat jer, uz minimalnu upotrebu koda, kreira očekivan gotov proizvod. Dakako da postoji puno mjesta za napredak, ali sam po sebi može se ocijeniti prolaznom ocjenom. Od verzije 11 omogućeno je i kreiranje web aplikacija što bi moglo pružiti novu priliku samom softveru za probijanje na tržište RAD alata.

## 5.2. Oracle APEX

Oracle Application Express (APEX) nastao je 2004. godine te je do danas razvijeno 22 stabilne verzije softvera. Od početka, APEX je baziran na kreiranju web aplikacija.

Ono što doista izdvaja ovaj alat je jednostavnost korištenja. Korisnik upoznat sa sučeljem te bazičnim znanjem PL/SQL jezika može razviti web aplikaciju sa gotovo svim funkcionalnostima koje mu mogu biti potrebne. Upravo zato, APEX je prilično popularan i kod studenata koji kroz korištenje softvera mogu pokazati svo znanje i sposobnost u kreiranju poslovnih aplikacija.

APEX je u potpunosti besplatan, omogućuje procesiranje i validaciju na strani servera, pruža temeljnu podršku za grupno razvijanje aplikacija te nudi (od strane Oracle-a) besplatno *hostanje* demo verzija aplikacije.

Ono što se može izdvojiti kao negativna strana je ograničenost alata za kreiranje kao i baza podataka samo na one koje pruža Oracle. Također, mali broj poslužitelja prihvaća APEX (Oracle bazu podataka) pa je potrebno instalirati potrebne pakete na operacijskom sustavu kako bi aplikacija radila na besplatnoj Oracle Express verziji baze podataka (Oracle XE).

Tijekom izrade ove aplikacije, sustav se pokazao iznad svih očekivanja. Gotovo svaku ideju bilo je moguće realizirati te implementirati u aplikaciju. Uz pomoć APEX-a, web aplikacija pruža toliko veći broj funkcionalnosti da ju je gotovo nemoguće povezati s *desktop* aplikacijom na temelju koje je kreirana.

## 6. Zaključak

Svrha ovog rada bila je prikazati na razumljiv i sveobuhvatan način kreiranje jedne poslovne web aplikacije. Prvotno kroz teorijsku pozadinu, a onda i korištenje samog RAD alata ideja je bila predočiti sve korake kreiranja baze podataka.

Željelo se naglasiti i važnost unapređenja postojeće *desktop* aplikacije u web oblik. Kako je rečeno i u uvodu, upravo zbog implementiranosti Interneta u sve aspekte modernog života, ovakve aplikacije predstavljaju presudnu komponentu svakog uspješnog poslovanja. Iz toga izlazi i važnost kvalitete takvog softvera. Upravo prisutnost na World Wide Webu uz sve prednosti, pruža i veću dozu opasnosti za sigurnost, integritet i stabilnost podataka. Zato je vrlo važna sigurnost koju pružatelj usluge nudi.

Sama aplikacija kreirana je na način da pokrije sve nedostatke i potrebe temeljne *desktop* verzije. Iako je obujam podataka povećan dodavanjem tablica, izračuni su unaprijeđeni na način da predstavljaju realniju sliku te je dodan veliki broj dinamičkih akcija i izračuna, uvijek ima mjesta za napredak. Upravo zato i APEX označava ovu aplikaciju kao *version 1.0*. Kada bi se radilo o stvarnoj aplikaciji, developeri bi nastavili istraživati te dodavati funkcionalnosti u budućim zakrpama i verzijama.

Alat Oracle APEX opravdao je popularnost koju drži među studentima. Pokazao se kao softver visoke kvalitete s velikim brojem funkcionalnosti i mogućnosti koje su uvelike olakšale realiziranje ideja kao i općenito kreiranje aplikacije. Također, primjetna je bila i razlika u vremenu uloženom u izradu aplikacije, koje je bilo znatno kraće nego kod kreiranja temeljne aplikacije u Clarionu. Uz intuitivno i jednostavno sučelje te spomenuti velik broj funkcionalnosti, može se reći da je bio užitak koristiti ovaj alat.

Obzirom da je model i dijagram baze bio praktički gotov tj. tražio je samo nekoliko modifikacija, proces izrade išao je poprilično glatko. Kad se počelo s izradom same aplikacije, prijašnje znanje i iskustvo te APEX-ova elegancija omogućili su brzo i efikasno kreiranje zamišljene aplikacije.

Konačno, bitno je naglasiti važnost ovakvog zadatka za učenje i razvijanje. Iako se radilo o korištenju poznatog softvera, sama izrada pružila je mnogo prilika za učenje novih mogućnosti i funkcionalnosti te razvijanje vještina. Obzirom na važnost poslovnih aplikacija te spomenuti aspekt učenja može se reći da ovakav zadatak nema samo akademsku, već i osobnu važnost.

## Literatura

- [1] Štorić, L. (2019) *Poslovna aplikacija za dostavnu službu nad relacijskom bazom podataka – Clarion*. Završni rad. Rijeka: Sveučilište u Rijeci.
- [2] E. F. Codd. 1990. The relational model for database management: version 2. Addison-Wesley Longman Publishing Co., Inc., USA, 1990.godina
- [3] Pavlić Mile, Oblikovanje baze podataka, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2011.godina

## Popis slika

SLIKA 1. DIJAGRAM ENTITETI-VEZE .....	8
SLIKA 2. <i>APP BUILDER</i> .....	10
SLIKA 3. <i>BLUEPRINT</i> APLIKACIJE.....	11
SLIKA 4. GLAVNI ZASLON .....	11
SLIKA 5. <i>OBJECT BROWSER</i> .....	12
SLIKA 6. NOVA TABLICA .....	12
SLIKA 7. POSTAVLJANJE PRIMARNOG KLJUČA .....	13
SLIKA 8. POSTAVLJANJE VANJSKOG KLJUČA .....	14
SLIKA 9. POSTAVLJANJE OGRANIČENJA.....	14
SLIKA 10. PRIMJER KLJUČEVA KOD AGREGACIJE .....	15
SLIKA 11. <i>MANAGE USERS AND GROUPS</i> .....	16
SLIKA 12. KREIRANJE KORISNIČKOG RAČUNA .....	16
SLIKA 13. <i>REPORT WITH FORM</i> .....	17
SLIKA 14. <i>REPORT WITH FORM</i> - OSNOVNE POSTAVKE .....	18
SLIKA 15. <i>PAGE DESIGNER</i> .....	19
SLIKA 16. POPIS DOSTAVNIH VOZILA .....	20
SLIKA 17. <i>PAGE DESIGNER</i> - UNOS DOSTAVNIH VOZILA .....	20
SLIKA 18. <i>VALIDATION - MAXIMUM LENGTH</i> .....	21
SLIKA 19. <i>SELECT LIST - STATIC VALUES</i> .....	22
SLIKA 20. KREIRANJE <i>MASTER DETAIL</i> STRANICE - <i>MASTER</i> TABLICA.....	23
SLIKA 21. KREIRANJE <i>MASTER DETAIL</i> STRANICE - <i>DETAIL</i> TABLICA .....	23
SLIKA 22. POPIS EVIDENCIJSKIH LISTOVA .....	25
SLIKA 23. POPIS EVIDENCIJSKIH LISTOVA – DOSTAVLJAČ.....	26
SLIKA 24. UNOS DOSTAVE .....	27
SLIKA 25. OSNOVICE DOSTAVLJAČA.....	28
SLIKA 26. UNOS PLAĆE DOSTAVLJAČA - DINAMIČKE AKCIJE .....	30
SLIKA 27. METRIKE .....	32
SLIKA 28. POČETNA STRANICA – ADMINISTRATOR .....	34
SLIKA 29. POČETNA STRANICA - DOSTAVLJAČ .....	35
SLIKA 30. FEEDBACK .....	36
SLIKA 31. STRANICA LOGIN.....	36
SLIKA 32. <i>THEME ROLLER</i> .....	37
SLIKA 33. GRAFIČKI DIZAJN - CLARION I APEX.....	38

# Prilozi

## Zadatak za Diplomski rad



Rijeka, 10. lipnja 2021.

### Zadatak za diplomski rad

Pristupnik: Luka Štorić

Naziv diplomskog rada: Izrada poslovne web aplikacije u Oracle APEX 20.1

Naziv diplomskog rada na eng. jeziku: Development of a web-based business application in Oracle APEX 20.1

Sadržaj zadatka:

Zadatak diplomskog rada je opisati postupak izrade poslovne web aplikacije u alatu Oracle Application Express (Oracle APEX) za dostavnu službu. Cilj je prikazati način korištenja alata Oracle APEX kroz vlastito osmišljeni primjer, gdje će sama izrada aplikacije biti popraćena detaljnim objašnjenjima i slikama ekrana. Prije same izrade aplikacije potrebno je opisati poslovni proces dostavne službe, izraditi dijagram entiteta - veze te relacijski model. Na kraju rada poželjno je dati usporedbu Oracle Apex i Clarion alata.

Mentor:

Prof. dr.sc. Patrizia Pošćić

Voditeljica za diplomske radove:

Izv. prof. dr. sc. Ana Meštrović

Komentor:

Zadatak preuzet: 10. lipnja 2021.

(potpis pristupnika)