

Poslovna aplikacija u Oracle Apexu za rezervaciju smještaja

Sirotić Vujnović, Maria

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:195:359492>

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

Download date / Datum preuzimanja: **2024-05-19**



Sveučilište u Rijeci
**Fakultet informatike
i digitalnih tehnologija**

Repository / Repozitorij:

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



Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Maria Sirotić Vujnović

Poslovna aplikacija u Oracle Apexu za rezervaciju smještaja

Završni rad

Mentor: doc.dr.sc. Martina Ašenbrener Katić

Rijeka, rujan 2023.

Rijeka, 12. lipnja 2023.

Zadatak za završni rad

Pristupnica: Maria Sirotić Vujnović

Naziv završnog rada: Poslovna aplikacija u Oracle Apexu za rezervaciju smještaja

Naziv završnog rada na engleskom jeziku: Business application in Oracle Apex for booking accommodations

Sadržaj zadatka:

Zadatak završnog rada je opisati postupak planiranja i stvaranja aplikacije za rezervaciju smještaja. Procesu izrade aplikacije prethodit će analiza sustava, definiranje korisničkih zahtjeva i modela korisnika te oblikovanje baze podataka. Potrebno je aplikaciju izraditi koristeći Oracle Apex te opisati postupak izrade aplikacije..

Mentor

Doc. dr. sc. Martina Ašenbrener Katić



Voditelj za završne radove

Doc. dr. sc. Miran Pobar



Zadatak preuzet: 19. lipnja 2023.



(potpis pristupnice)

Sažetak

U ovom radu detaljno je prikazan i opisan proces razvoja poslovne aplikacije za rezervaciju smještaja, koristeći se alatom Oracle Apex. Glavni cilj rada je demonstrirati korištenje navedenog alata kroz stvarni primjer razvoja vlastite aplikacije, popraćen detaljnim opisima i snimaka ekrana prilikom postupka izrade iste. Prije izrade same aplikacije, kreirani su modeli korisnika i korisničke priče, kako bi razvoj bio usmjeren prema korisnicima. Potom je izrađen dijagram entiteta i veza prema metodologiji MIRIS (koristeći Microsoftov program Visio) koji je potom preveden u relacijski model, a temeljem kojega je kreirana konačna relacijska baza podataka. Naposljetku je izrađena aplikacija nad tom bazom podataka, a koja podržava ključne operacije pregledavanja, dodavanja, ažuriranja i brisanja podataka iz baze podataka, poznate kao CRUD operacije (*Create, Read, Update, Delete*). Prikazan je i konačan izgled aplikacije kao i opisano korištenje iste.

Ključne riječi

Poslovna aplikacija, Oracle APEX, Rezervacija smještaja, Modeli korisnika, Dijagram, DEV, Relacijski model, CRUD, Primarni ključ, Vanjski ključ, Baza podataka, Relacija, Atribut, Entitet

Sadržaj

1. Uvod	5
2. Korisničke priče i modeli korisnika	7
2.1. Korisničke priče	7
2.2. Modeli korisnika.....	8
3. Model entiteti-veze.....	10
4. Relacijski model	16
5. Izrada aplikacije	17
5.1. Postavljanje radnog okruženja.....	17
5.2. Kreiranje baze podataka – SQL workshop	20
5.3. Kreiranje aplikacije	28
5.4. Regije statičnog sadržaja i klasična izvješća	31
5.5. Master Detail izvješće	38
5.6. Interaktivne mreže (Interactive Grids)	42
5.6.1. Rezervacije i plaćanja.....	42
5.6.2. Države i mjesta	47
5.7. Forme za unos podataka (Form).....	47
5.8. Interaktivna izvješća (Interactive Reports).....	52
5.9. Validacija podataka	53
5.10. Autentifikacija i autorizacija korisnika	55
5.10.1. Autentifikacija	55
5.10.2. Autorizacija	61
6. Prikaz kreirane aplikacije	67
7. Zaključak	71
8. Popis literature.....	72
9. Popis slika	73
10. Popis priloga.....	76

1. Uvod

Sa sve većom potražnjom za brzim i jednostavnim načinima rezervacije smještaja, digitalne poslovne aplikacije postaju neprocjenjiv saveznik u olakšavanju ovog procesa. Upravo u tom kontekstu, ovaj rad usmjeren je na razvoj poslovne aplikacije koja će znatno unaprijediti postupak rezervacije smještaja. Glavni cilj rada je detaljno prikazati i opisati korake u razvoju navedene aplikacije, korak po korak, koristeći Oracle APEX, kako bi se na konkretnom primjeru ilustrirale brojne mogućnosti ovoga alata.

Web aplikacija za rezervaciju smještaja svojim će korisnicima nuditi mogućnosti brzog i jednostavnog pronalaženja i rezervacije smještajnih jedinica kao što su hoteli, hosteli, apartmani te kuće za odmor. Dostupne smještaje korisnici će moći filtrirati prema nekolicini kriterija, primjerice lokaciji ili vrsti smještaja kojeg se želi rezervirati. Nakon što pronađe i odabere željenu smještajnu jedinicu, korisnik vrši rezervaciju na način da unese sve potrebne podatke koji se od njega traže. Korisnici će se u aplikaciju moći prijaviti sa svojim korisničkim podacima te time dobiti opciju pregleda vlastitih rezervacija, kako trenutnih tako i prošlih. Cilj same aplikacije jest olakšati korisnicima pronalazak i rezervaciju smještaja po njihovim vlastitim željama i potrebama uz garantiranu sigurnost i jednostavnost tog procesa.

Glavne funkcionalnosti aplikacije uključuju:

- Pretraživanje smještaja: korisnici mogu pretraživati smještajne jedinice prema različitim kriterijima poput lokacije i vrste smještaja te pregledati detalje o konkretnom smještaju i njegovim cijenama.
- Rezervaciju smještaja: korisnici mogu odabrati željeni smještaj i unijeti podatke za rezervaciju kao što su osobni podaci, datum prijave i odjave, način plaćanja, odnosno sve potrebne informacije koje aplikacija od njih traži.
- Pregled rezervacija: korisnici imaju uvid u sve svoje prošle i sadašnje rezervacije ostvarene putem aplikacije te pregled detalja i plaćanja o istima.
- Autentifikaciju i autorizaciju korisnika: korisnici imaju mogućnost prijave u aplikaciju putem svojih korisničkih podataka, a različitim vrstama korisnika dostupne su različite funkcionalnosti unutar same aplikacije.

Kao što je već navedeno, za izradu aplikacije korišten je Oracle APEX (*Oracle Application Express*), specifično verzija 23.1. Oracle APEX je visokoproduktivna platforma za razvoj web i mobilnih aplikacija koja omogućuje brzo stvaranje skalabilnih i sigurnih aplikacija putem grafičkog korisničkog sučelja, s minimalnim potrebama za kodiranjem. Koristeći APEX, programeri i developeri mogu brzo razviti i implementirati privlačne aplikacije koje rješavaju stvarne probleme. Ključna prednost APEX-a leži u minimalnim zahtjevima za kodiranjem, čineći ga pristupačnim širem korisničkom krugu, ne samo IT stručnjacima. Nije potrebno imati napredno tehnološko znanje kako bi se kreirala i isporučila sofisticirana rješenja pomoću ovog alata. Ovaj se alat može koristiti kao Oracle Cloud usluga putem web preglednika ili se može preuzeti na računalo za lokalni razvoj aplikacija (Oracle, 2023.).

Oracle APEX pripada kategoriji alata za brzi razvoj aplikacija, tzv. RAD alata (*Rapid Application Development tools*) koji značajno pojednostavljaju proces razvoja aplikacija kroz

grafička sučelja i generiranje dijela programskog koda. Važno je napomenuti da, iako APEX može automatski generirati dio koda, i dalje je potrebno određeno stručno znanje za optimalan dizajn i funkcionalnost aplikacija. RAD alati, uključujući APEX, omogućuju programerima i developerima da brže razvijaju aplikacije, posebno u usporedbi s tradicionalnim metodama razvoja. Kroz upotrebu grafičkih sučelja, predložaka i drugih alata, RAD pristup olakšava implementaciju funkcionalnosti i smanjuje potrebu za dugotrajnim ručnim kodiranjem, čime se veći fokus pridaje stvaranju inovativnih rješenja i rješavanju stvarnih problema.

2. Korisničke priče i modeli korisnika

2.1. Korisničke priče

Korisničke priče (eng. *User stories*) su koncept koji pomaže u analizi i usredotočuje se na ciljeve krajnjih korisnika, odnosno na funkcionalne zahtjeve zapisane iz perspektive korisnika. Korisnička je priča u biti jednostavan opis onoga što će softver moći obavljati, odnosno zapis značajki softvera na prirodnom jeziku, ispričan iz perspektive korisnika ili kupca. Ona opisuje vrstu korisnika, što on želi postići i zašto, omogućujući developerima da razumiju što je sve potrebno implementirati i razviti kako bi zadovoljili potrebe korisnika (Čandrlić, 2023.).

Korisničke priče slijede određeni format koji uključuje tri ključna elementa (Čandrlić, 2023.):

1. Korisnik (uloga) – opisuje tko je korisnik ili koja uloga korisnika (osoba, sustav, podsustav) ima interes za određenom funkcionalnošću, odnosno predstavlja subjekt koji će komunicirati sa sastavom (npr. aplikacijom) koji treba biti implementiran
2. Akcija – definira što korisnik želi postići ili što želi da aplikacija učini za njega, tj. predstavlja korisničko očekivanje koje se može ostvariti interakcijom sa sustavom
3. Korist – opisuje koju korist korisnik dobiva iz ispunjenja ove funkcionalnosti, odnosno to je vrijednost koja proizlazi iz interakcije sa sustavom

Iz rečenoga slijedi da se prilikom zapisivanja korisničkih priča vodimo sljedećim predloškom:

Kao [vrsta korisnika – tko] želim [neki cilj – što] tako da [neki razlog – zašto].

Korisničke priče vezane uz razvoj aplikacije za rezervaciju smještaja su sljedeće:

- Kao korisnik, želim pregledati dostupne smještaje i njihove cijene kako bih mogao pronaći idealni smještaj za svoj boravak.
- Kao korisnik, želim pretraživati smještaje na temelju različitih kriterija, poput lokacije, i vrste smještaja kako bih pronašao smještaj koji odgovara mojim željama.
- Kao korisnik, želim izvršiti rezervaciju smještaja, odabrati datume boravka i unijeti podatke o plaćanju.
- Kao korisnik, želim pregledati sve moje trenutne i prethodne rezervacije kako bih vidio datume boravka, cijene i ostale pojedinosti konkretne rezervacije.
- Kao korisnik, želim pronaći smještaj u kojem je moguć boravak određenog broja osoba kako bih pronašao najbolji smještaj za grupu ljudi s kojima putujem.
- Kao administrator, želim pregledati sve dostupne smještaje, dodavati nove smještaje kao i ažurirati ili brisati one postojeće kako bih aplikaciju održavao ažurnom.
- Kao administrator, želim imati mogućnost upravljanja korisnicima te obavljanja autentifikacijskih i autorizacijskih postupaka kako bih osigurao sigurnost aplikacije.

2.2. Modeli korisnika

Modeli korisnika ili korisničke persone (eng. *User personas*), su arhetipovi korisnika (apstraktne reprezentacije različitih skupina ponašanja) nastali grupiranjem podataka o ponašanju, stavovima, ciljevima, preferencijama i motivaciji korisnika, a za koje se smatra da će koristiti uslugu koja se razvija. One ne predstavljaju stvarne ljude već fiktivne likove – modele osoba kreirane na temelju prikupljenih podataka o ponašanju i motivaciji stvarnih korisnika (Čandrlić, 2023.).

Kreiranje modela korisnika omogućuje bolje razumijevanje potreba, iskustava, ponašanja i ciljeva krajnjih korisnika, s obzirom da različiti korisnici imaju različita očekivanja i potrebe. Persone čine zadatak dizajniranja manje složenim, vode proces generiranja ideja i mogu značajno pomoći u stvaranju optimalnog korisničkog iskustva za ciljanu skupinu korisnika. Kreiranjem ovih fiktivnih likova, moguće je zamisliti kako bi različiti korisnici mogli koristiti određeni proizvod ili uslugu, što pomaže developerima da bolje zadovolje stvarne potrebe korisnika, čime se postiže bolji korisnički doživljaj i veći uspjeh na tržištu.

Očekivane grupe korisnika koje će koristiti aplikaciju za rezervaciju smještaja koja se razvija su:

- Turisti
- Studenti
- Obitelji / parovi
- Ljudi koji putuju poslovno

Kreirani modeli korisnika prikazani su slikama 1., 2. i 3., a izrađeni su koristeći web alat *Xtensio* (dostupno na: <https://xtensio.com/>).

Ana Ivanović



Dob: **40**
Mjesto: **Rijeka**
Zanimanje: **Voditeljica marketinga**
Obitelj: **Suprug i dvoje djece**

"Success is not final; failure is not fatal: it is the courage to continue that counts."
Winston Churchill

Biografija

Ana je poslovna žena koja putuje na različita mjesta u zemlji zbog svojeg posla. Često putuje sama pa se trudi pronaći smještaj koji je udoban, siguran i ugodan. Iako najviše voli boraviti u hotelima, ona također razmatra druge mogućnosti smještaja poput apartmana i hostela. Ana je otvorena prema različitim vrstama smještaja i uvijek se trudi pronaći onaj koji najbolje odgovara njezinim potrebama i željama. Ana je svjesna važnosti putovanja i uvijek se trudi stvoriti ravnotežu između poslovnog i privatnog života tijekom putovanja

Veza s tehnologijom

Ana je dobar poznavatelj moderne tehnologije, obzirom da se s istom susreće svakodnevno na radom mjestu. Osim u poslovne svrhe, tehnologijom se uvelike koristi i u privatnom životu te posjeduje prijenosno računalo, pametni sat i pametni telefon.

Krajnji cilj

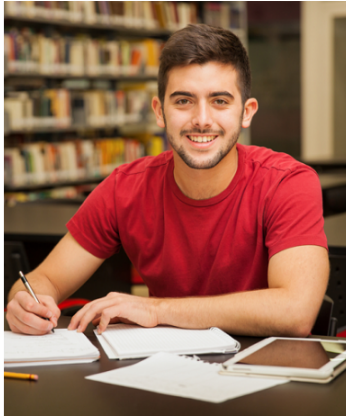
- Pronaći smještaj za službeno putovanje

Scenariji korištenja

- Želi brzo i jednostavno pronaći smještaj za svoje putovanje
- Želi brzo rezervirati smještaj uz pomoć jednostavnog odabira načina plaćanja
- Želi pregledati sve svoje trenutne i prethodne rezervacije na jednom mjestu

Slika 1. Model prvog korisnika

Marko Radić



Dob: **22**
Mjesto: **Osijek**
Zanimanje: **Student**
Obitelj: **Roditelji i sestra**

"The world is a book, and those who do not travel read only one page."
Saint Augustine

Biografija

Marko je student na Građevinskom i arhitektonskom fakultetu u Osijeku koji voli putovati tijekom pauza između semestara. Često putuje solo ili s prijateljima i traži pristupačan smještaj koji će mu omogućiti uštedu novca. Spreman je spavati u hostelu ili dijeliti sobu kako bi uštedio novac te preferira skromniji smještaj kako bi ostatak budžeta potrošio na ostale aktivnosti. Marko je zapravo vrlo pustolovan student kojemu je cilj upoznati razne kulture te posjetiti mnoge destinacije, no uzimajući u obzir pritom svoje novčane mogućnosti njega kao studenta.

Veza s tehnologijom

S obzirom na dob, Marku je tehnologija dio svakodnevice, bilo da se njome služi u svrhu obrazovanja i istraživanja, bilo u privatne svrhe i za zabavu. Osim toga, veliki je fan i društvenih mreža putem kojih voli svojim prijateljima podijeliti slike sa mnogobrojnih putovanja na koja ide.

Krajnji cilj

- Pronaći jeftin smještaj za kratko putovanje

Scenariji korištenja

- Želi pronaći najjeftiniji smještaj koji odgovara njegovim putnim planovima i budžetu
- Želi pregledati više opcija smještaja kako bi rezervirao onaj koji mu najviše odgovara
- Želi provjeriti dostupnost smještaja na datume kada putuje

Slika 2. Model drugog korisnika

Ivan Kružić



Dob: **54**
Mjesto: **Karlovac**
Zanimanje: **Vlasnik lanca restorana**
Obitelj: **Supruga i kćer**

"Traveling is like flirting with life. It's like saying, 'I would stay and love you, but I have to go; this is my station.'"
Lisa St. Aubin de Teran

Biografija

Ivan je osoba srednje životne dobi koja je cijeli život posvetila poslu, a sada želi putovati sa svojom ženom i uživati u svojim hobijima - putovanju i fotografiranju. Nakon godina rada kao vlasnik lanca restorana, Ivan je odlučio prepustiti svoj posao kćeri i konačno se posvetiti sebi i svojim željama. Ivan zajedno sa suprugom želi putovati diljem svijeta i istraživati nove kulture, okuse i prizore, a traže i savršeno mjesto za odmor nakon uzbudljivog dana. Upravo zato, oni sada traže smještaj koji će im pružiti ugodan i udoban boravak tijekom njihovih putovanja te pružiti najbolji omjer kvalitete i cijene smještaja, a posebice su mu dragi apartmani, dok njegova supruga preferira kuće za odmor.

Veza s tehnologijom

Ivan nije preveliki fan tehnologije, no tokom života morao se njome koristiti i to pretežito u poslovne svrhe. Na računalu se dobro snalazi te kaže da je za njegove potrebe to i više nego dovoljno.

Krajnji cilj

- Ubrzati proces pronalaska i rezervacije smještaja s obzirom na česta putovanja.

Scenariji korištenja







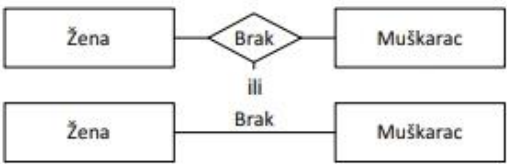


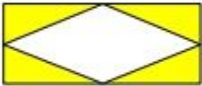



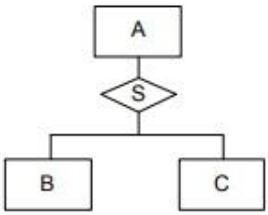
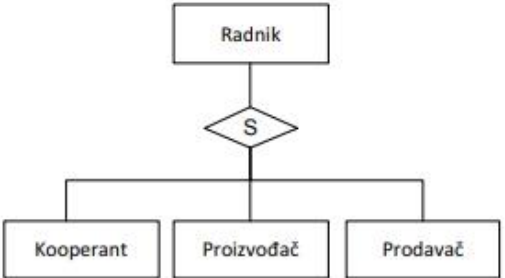
- Želi koristiti aplikaciju za brzo i jednostavno pronalaženje smještaja prema lokaciji, vrsti smještaja te njegovoj cijeni
- Želi jednostavan i efikasan proces rezervacije
- Želi dobiti detalje o konkretnom smještaju
- Želi pronaći smještaj prikladan za boravak dviju osoba u istome
- Želi pregled svih svojih trenutnih rezervacija

Slika 3. Model trećeg korisnika

3. Model entiteti-veze

Metoda entiteti-veze (skraćeno EV, eng. *Entity-relationship model*) jest grafički način prikaza grupa podataka koje su međusobno povezane u nekom promatranom sustavu. EV je semantički bogata metoda za modeliranje podataka i kroz niz varijanti polazne metode postala je jedna od najčešće korištenih metoda. Tako se EV koristi u specijaliziranim metodologijama, od kojih je jedna i MIRIS (skraćeno od hrv. Metodologija za Razvoj Informacijskog Sustava) – korištena i prilikom modeliranja nama potrebnog modela podataka.

Modeliranjem metodom entiteti-veze nastaje model, odnosno dijagram entiteta-veza, a čija kratica glasi DEV (engl. *Entity - Relationship Diagram, ERD*). Dakle, DEV je grafički prikaz modela podataka sustava, metodom entiteti-veze, pri čemu se svaki koncept dijagrama prikazuje sebi svojstvenim oblikom, kako je prikazano na slici 4.

KONCEPT	SIMBOL	PRIMJER
TIP ENTITETA		
SLAB TIP ENTITETA		
TIP VEZE	 ili 	
ATRIBUT		
AGREGACIJA		
POVRATNA VEZA		
GENERALIZACIJA		

Slika 4. Koncepti modela entiteti-veze (Pavlić, 2011.)

Za modeliranje dijagrama entiteta i veza, koji je potreban za aplikaciju rezervacije smještaja, korišteni su ključni koncepti poput entiteta, slabih tipova entiteta, agregacija, veza i atributa. Ovi koncepti su temeljni za izgradnju strukture DEV-a i njihova uloga je objašnjena u nastavku.

Entitet (engl. *Entity*) u sustavu predstavlja stvarni ili konceptualni element, koji se jasno razlikuje od drugih entiteta. Drugim riječima, to je pojam koji postoji u poslovnom okruženju i o kome se zahtijeva čuvanje određenih podataka.

Tip entiteta (engl. *Entity type*, skraćeno TE) je skup sličnih pojedinačnih entiteta dobiven klasifikacijskom apstrakcijom u kojoj se slični entiteti proglašavaju zajedničkim skupom nazvanim upravo Tip entiteta, odnosno TE je zapravo skup entiteta istoga tipa.

Jak tip entiteta (ili jednostavno Tip entiteta, engl. *Strong entity*) predstavlja tip entiteta koji posjeduje vlastiti primarni ključ te nije ovisan o drugim tipovima entiteta prisutnim u modelu podataka. Ovaj tip entiteta ima dovoljno informacija da ga se samostalno identificira i razlikuje od drugih entiteta, bez potrebe za povezivanjem s drugim entitetima kako bi bio jedinstven. Grafički je predstavljen pravokutnikom proizvoljnih dimenzija.

Slab tip entiteta (engl. *Weak entity*) je tip entiteta koji u određenom smislu ovisi o drugom tipu entiteta, pri čemu se ta ovisnost prikazuje kroz specijalni tip veze između jakog i slabog tipa entiteta. Grafički, tip slabog entiteta se prikazuje kroz dvostruki pravokutnik, s jednim pravokutnikom unutar drugoga. Specijalni tip veze grafički se prikazuje romбом ili linijom sa strelicom koja ide od jakoga ka slabom tipu entiteta .

Veza ili asocijacija (engl. *Relationship*) je koncept koji predstavlja interakciju ili odnos među entitetima u sustavu, pružajući informacije o njihovoj povezanosti. Veza predstavlja odnos koji među entitetima postoji bilo u stvarnom svijetu ili samo u mislima. Dva različita tipa entiteta mogu biti povezana putem više pojedinačnih veza, što nas dovodi do pojma Tip veze.

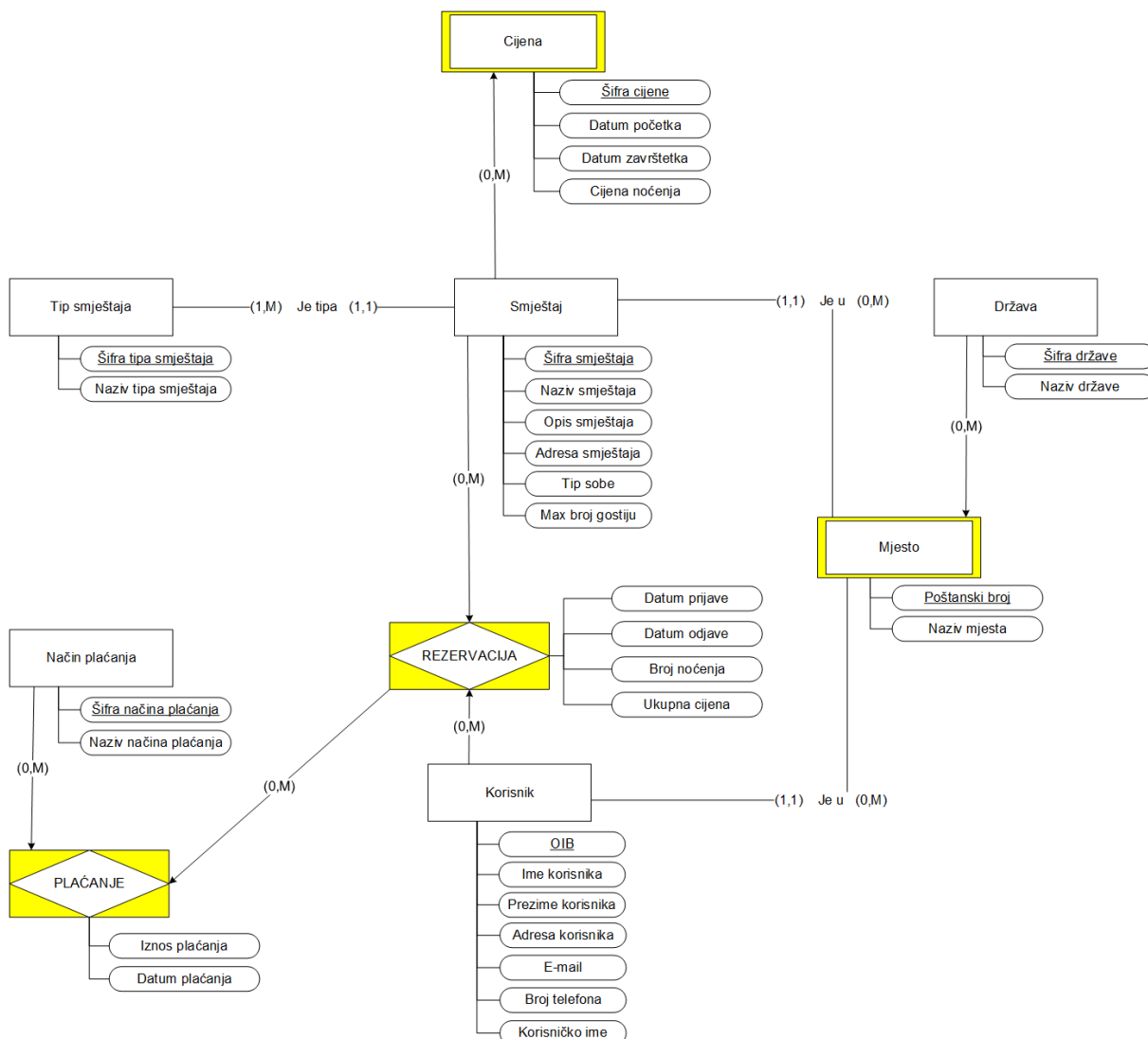
Tip veze (engl. *Relationship type*) je skup veza koje postoje između istih tipova entiteta. To je kolekcija pojedinačnih veza slične prirode. Grafički, tip veze prikazuje se romбом ili linijom proizvoljne veličine, a naziv tipa veze upisuje se unutar romba ili na liniju, kako bi jasno identificirao vrstu odnosa koji se modelira. Naziv tipa veze jest riječ ili skupina riječi koja predstavlja odnos koji postoji među entitetima. Brojnost tipa veze eksplicitno se označava na DEV-u i kaže koliko entiteta pojedinoga tipa entiteta E1 sudjeluje u tipu veze V s entitetom iz tipa entiteta E. U nastavku teksta, tip veze će se kraće nazivati samo „veza“, radi jednostavnosti pisanja.

Atribut entiteta (engl. *Attribute*) je neko svojstvo, neka imenovana karakteristika, odlika entiteta. Atribut omogućava interpretaciju tipa vrijednosti kroz pojavljivanje entiteta u tipu entiteta, čime prevodi podatak u informaciju. Dakle, atribut se definira kao povezivanje između entiteta tipa entiteta i vrijednosti tipa vrijednosti. Atribut možemo grafički prikazati kao oval unutar kojega je upisano ime samog atributa, a potrebno je jedinstveno imenovati sve attribute u modelu podataka.

Agregacija (također poznata kao mješoviti tip entiteta ili agregirani tip entiteta, eng. *Aggregation*) je apstrakcija u kojoj se tip veze između dvaju ili više tipova entiteta tretira kao novi tip entiteta. Ključ agregacije sastoji se od primarnih ključeva komponenata koje čine tu agregaciju. Grafički se prikazuje pravokutnikom u koji je ucrtan romb, a od tipova entiteta koji

čine agregaciju se povlači strelica prema simbolu agregacije, kako bi se naznačilo od kojih je tipova entiteta nastala agregacija.

Na slici 5. prikazan je DEV modeliran za poslovnu aplikaciju za rezervaciju smještaja, a u nastavku slijedi detaljan opis istoga.



Slika 5. Prikaz dijagrama entiteta-veza za aplikaciju za rezervaciju smještaja

Jaki tipovi entiteta sa prethodnog dijagrama su: TIP SMJEŠTAJA, SMJEŠTAJ, DRŽAVA, NAČIN PLAĆANJA i KORISNIK. U slabe tipove entiteta pripadaju CIJENA i MJESTO te su prisutne i dvije agregacije: REZERVACIJA i PLAĆANJE.

Svakome od navedenih tipova entiteta pridruženi su i njemu svojstveni atributi, a svaki od njih također posjeduje svoj primarni ključ, sukladno opisu koji slijedi. Kod jakog tipa entiteta jedan od atributa tog tipa entiteta odabran je kao primarni ključ (na DEV-u je taj atribut podvučen kako bi se jasno razlikovao od ostalih atributa). Slab tip entiteta posjeduje primarni ključ koji je kombinacija primarnog ključa jakog tipa entiteta i primarnog ključa slabog tipa entiteta. Agregacija nema svoj vlastiti primarni ključ; umjesto toga, primarni ključ agregacije kombinacija je primarnih ključeva tipova entiteta koji sudjeluju u toj agregaciji.

Jak tip entiteta DRŽAVA predstavlja različite države i pohranjuje podatke njihovim šiframa i nazivima, putem atributa *Šifra_države* i *Naziv_države*, respektivno. Atribut *Šifra_države* primarni je ključ ovog tipa entiteta. Ovaj tip entiteta povezan je sa slabim tipom entiteta MJESTO, što je ujedno i jedina veza u kojoj ovaj tip entiteta sudjeluje. S obzirom da je riječ o specijalnoj vezi, brojnost na strani tipa entiteta DRŽAVA iznosi (0,M), a na strani tipa entiteta MJESTO iznosi (1,1), što je standardna brojnost za specijalne tipove veza. Prema tome, zaključujemo da unutar jedne države može postojati nijedno ili mnogo mjesta, dok jedno konkretno mjesto može pripadati samo jednoj određenoj državi. Dodatni detalji o ovoj vezi dani su kod opisa slabog tipa entiteta MJESTO, odnosno u nastavku teksta koji slijedi.

Slab tip entiteta MJESTO opisuje konkretna mjesta unutar određenih država, omogućavajući time klasifikaciju smještaja i korisnika prema mjestu (a time i prema državi). Pripadni atributi ovog tipa entiteta su *Poštanski_broj* i *Naziv_mjesta*, pri čemu je *Poštanski_broj* dio primarnog ključa ovog tipa entiteta. Ovaj je tip entiteta slab u odnosu na tip entiteta DRŽAVA, što možemo zaključiti iz načina na koji je nacrtana veza između ovih dvaju tipova entiteta (dodatak strelice na strani slabog tipa entiteta). Iz navedenoga slijedi da je primarni ključ ovog tipa entiteta sastavljen od atributa *Šifra_države* i *Poštanski_broj*, kako bismo jedinstveno mogli identificirati sva mjesta u svim državama. Nadalje, ovaj tip entiteta sudjeluje u vezama s dvama jakim tipovima entiteta (SMJEŠTAJ i KORISNIK), pri čemu je u oba slučaja brojnost na strani tipa entiteta MJESTO definirana kao (0,M), a na strani tipova entiteta SMJEŠTAJ i KORISNIK ona iznosi (1,1). Ovo implicira da se u jednom mjestu može nalaziti niti jedan ili više korisnika i smještaja, dok se jedan korisnik ili smještaj može nalaziti u jednom i samo jednom mjestu.

Jaki tip entiteta TIP SMJEŠTAJA kategorizira različite vrste smještaja koji se nude, kao što su hoteli, hosteli, apartmani i slično. Navedeni tip entiteta sadrži attribute *Šifra_tipa_smještaja* i *Naziv_tipa_smještaja*, od kojih je atribut *Šifra_tipa_smještaja* definiran kao primarni ključ. Ovaj tip entiteta povezan je s tipom entiteta SMJEŠTAJ putem veze koja ima brojnost (0,M) na strani tipa entiteta TIP SMJEŠTAJA, te brojnost (1,1) na strani tipa entiteta SMJEŠTAJ. Ova veza podrazumijeva da jedan tip smještaja može biti povezan s nulom ili više smještaja, odnosno da određeni tip smještaja može imati smještaje koji mu pripadaju, ali ne mora. S druge strane, brojnost (1,1) sa strane tipa entiteta SMJEŠTAJ implicira da svaki smještaj mora biti povezan s točno jednim tipom smještaja.

Jaki tip entiteta SMJEŠTAJ predstavlja sve konkretne smještajne objekte koje je moguće rezervirati te sadrži sve potrebne informacije o njima. Atributi koji pripadaju ovom tipu entiteta su *Šifra_smještaja*, *Naziv_smještaja*, *Opis_smještaja*, *Adresa_smještaja*, *Tip_sobe*, *Max_broj_gostiju*. Kao primarni ključ ovog tipa entiteta odabran je atribut *Šifra_smještaja*. Ovaj tip entiteta predstavlja jaku komponentu u odnosu na slabi tip entiteta CIJENA (na DEV-u označeno strelicom od tipa entiteta SMJEŠTAJ ka tipu entiteta CIJENA). Također, ovaj tip entiteta jedna je od komponenti koja tvori agregaciju REZERVACIJA. Detalji o oba navedena odnosa biti će naknadno opisani prilikom prezentiranja informacija o slabom tipu entiteta CIJENA te agregaciji REZERVACIJA. Posljednja veza ovog atributa – ona s tipom entiteta MJESTO – već je ranije opisana (svaki smještaj lociran je u jednom i samo jednom mjestu).

Slabi tip entiteta CIJENA definira cijene noćenja u određenom smještajnom objektu, pri čemu cijene variraju ovisno o vremenskom razdoblju. Pripadni atributi su *Šifra_cijene*,

Datum_početka, *Datum_završetka* i *Cijena_noćenja*. Kao što je već ranije napomenuto, ovaj je tip entiteta ovisan o jakom tipu entiteta SMJEŠTAJ, što implicira da atribut *Šifra_smještaja* također čini dio njegovog primarnog ključa. Stoga, cjeloviti primarni ključ ovog tipa entiteta čini kombinacija atributa *Šifra_smještaja* i *Šifra_cijene*.

Jaki tip entiteta KORISNIK predstavlja sve pojedince koji koriste aplikaciju za rezervaciju smještaja kako bi putem nje izvršili željenu rezervaciju. Atributi ovog tipa entiteta uključuju: *OIB*, *Ime_korisnika*, *Prezime_korisnika*, *Adresa_korisnika*, *E-mail*, *Broj_telefona* i *Korisničko_ime*. Primarni ključ za identifikaciju svih pojedinih korisnika jest atribut *OIB*. Kao što je prethodno navedeno, ovaj tip entiteta sudjeluje u vezi s tipom entiteta MJESTO, pri čemu se jedan korisnik nalazi (boravi) u najmanje jednom i najviše jednom mjestu. Također, tip entiteta KORISNIK uključen je i u agregaciju REZERVACIJA, koja je opisana u nastavku.

Agregacija REZERVACIJA omogućuje pohranu informacija o rezervacijama smještaja, povezujući smještaj s korisnikom koji je izvršio rezervaciju. Atributi ove agregacije su sljedeći: *Datum_prijave*, *Datum_odjave*, *Broj_noćenja* te *Ukupna_cijena*. U ovoj agregaciji sudjeluju dva jaka tipa entiteta – SMJEŠTAJ i KORISNIK. Agregacija sama po sebi nema vlastite attribute koji bi bili definirani kao primarni ključ. Umjesto toga, njen primarni ključ sastoji od primarnih ključeva komponenata koje ju tvore (tip entiteta SMJEŠTAJ i tip entiteta KORISNIK), konkretno od atributa *Šifra_smještaja* i *Šifra_korisnika*. Veze između jakih tipova entiteta i ove agregacije imaju brojnost (0,M) na strani jakih tipova entiteta, što je zapravo i razlog postojanja ove agregacije. U slučaju kada veza između dvaju tipova entiteta ima brojnost M po gornjoj granici s obje strane, takva se veza definira konceptom agregacije. U ovom slučaju, ukoliko bi između tipova entiteta SMJEŠTAJ i KORISNIK postojala „obična“ veza, brojnost s obje strane iznosila bi (0,M), a što nije dozvoljeno, te je tada vezu potrebno pretvoriti u agregaciju. Ovakva brojnost veza nam govori da je jedan smještaj moguće rezervirati niti jednom ili više puta te da jedan korisnik može izvršiti nula ili pak mnogo rezervacija. Navedeno nije moguće pratiti putem obične veze među tipovima entiteta. Dodavanjem agregacije možemo precizno pratiti koliko je puta korisnik izvršio rezervacije i obrnuto, koliko je puta jedan smještaj rezerviran od strane korisnika. Ova je agregacija dodano povezana s agregacijom PLAĆANJE, na način opisan u nastavku.

Jak tip entiteta NAČIN PLAĆANJA pohranjuje različite metode koje korisnici mogu koristiti prilikom plaćanja rezervacije, kao što su gotovinska i kartična plaćanja. Atributi ovog tipa entiteta su *Šifra_načina_plaćanja* i *Naziv_načina_plaćanja*, pri čemu je atribut *Šifra_načina_plaćanja* odabran kao primarni ključ putem kojeg će biti moguće jedinstveno identificiranje svih mogućih različitih načina plaćanja. Ovaj tip entiteta jedna je od komponenti koja čini agregaciju PLAĆANJE, te je to ujedno i jedina veza u kojoj ovaj tip entiteta sudjeluje. Brojnost ove veze sa strane tipa entiteta NAČIN PLAĆANJA iznosi (0,M) što implicira da se jedan način plaćanja veže uz niti jedno ili pak više plaćanja.

Agregacija PLAĆANJE pohranjuje podatke o transakcijama plaćanja koje su povezane s rezervacijama. Sadrži informacije o iznosu i datumu plaćanja te povezuje način plaćanja sa smještajem i korisnikom koji je izvršio plaćanje. U ovoj agregaciji sudjeluju tip entiteta NAČIN PLAĆANJA i agregacija REZERVACIJA. Atributi ove agregacije su *Iznos_plaćanja* i *Datum_plaćanja*, a primarni ključ sastoji se od primarnih ključeva tipova entiteta koji čine ovu

agregaciju, tj. od atributa *Šifra_smještaja*, *OIB* i *Šifra_načina_plaćanja*. Brojnost obiju veza ove agregacije iznosi (1,1) na strani agregacije, s obzirom da jedno plaćanje pripada samo jednom načinu plaćanja te samo jednoj konkretnoj rezervaciji. Putem ove agregacije omogućeno je praćenje svih plaćanja povezanih s jednom konkretnom rezervacijom, primjerice ukoliko je dio rezervacije plaćen gotovinom, a dio karticom. Bilježi se datum tih pojedinih plaćanja, kao i iznos plaćen prilikom svake transakcije.

4. Relacijski model

Pravila potrebna za prevođenje kreiranog dijagrama entiteti-veze su sljedeća (Pavlič, 2011.):

1. Atribut u DEV-u odgovara konceptu Atribut u RM-u.
2. Svaki tip entiteta iz DEV-a postaje relacija (tablica) u RM-u i to tako da atributi tipa entiteta postaju atributi relacije, primarni¹ (glavni) ključ tipa entiteta postaje primarni (glavni) ključ relacije.
3. Svaka agregacija postaje relacija u RM-u, ključ relacije je složen od ključeva tipova entiteta koji sudjeluju u vezi. Atributi agregacije postaju atributi relacije.
4. Svaki tip veze s brojnostima (1,1):(0,M) (tip veze gdje je strana s brojnošću po gornjoj granici 1 totalna) ne prevodi se u relaciju, već se ključ tipa entiteta koji ulazi u vezu sa strane s brojnošću M umeće kao atribut (vanjski ključ²) u relaciju dobivenu prevođenjem tipa entiteta s brojnošću (1,1).
5. Slabi tip entiteta prevodi se u relaciju. Atributi tipa slabog entiteta postaju atributi relacije. Ključ relacije je složen i od ključa jakoga tipa entiteta i ključa slaboga tipa entiteta tako da jedinstveno identificira n-torke relacije u okviru relacijskoga modela.

Na temelju modela entiteti-veze i pravila za prevođenje u relacijski model, dobiven je relacijski model prikaz niže u nastavku teksta. Pritom su primarni ključevi tipova entiteta prikazani podebljanim stilom, dok su vanjski ključevi nakošeni.

DRŽAVA (**Šifra_države**, Naziv_države)

MJESTO (**Šifra_države**, **Poštanski_broj**, Naziv_mjesta)

TIP SMJEŠTAJA (**Šifra_tipa_smještaja**, Naziv_tipa_smještaja)

SMJEŠTAJ (**Šifra_smještaja**, Naziv_smještaja, Opis_smještaja, Adresa_smještaja, Tip_sobe, Max_broj_gostiju, Šifra_tipa_smještaja, Šifra_države, Poštanski_broj)

CIJENA (**Šifra_smještaja**, **Šifra_cijene**, Datum_početka, Datum_završetka, Cijena_noćenja)

KORISNIK (**OIB**, Ime_korisnika, Prezime_korisnika, Adresa_korisnika, E-mail, Broj_telefona, Korisničko_ime, Šifra_države, Poštanski_broj)

NAČIN PLAĆANJA (**Šifra_načina_plaćanja**, Naziv_načina_plaćanja)

REZERVACIJA (**Šifra_smještaja**, **OIB**, Datum_prijave, Datum_odjave, Broj_noćenja, Ukupna cijena)

PLAĆANJE (**Šifra_smještaja**, **OIB**, **Šifra_načina_plaćanja**, Iznos_plaćanja, Datum_plaćanja)

¹ Primarni ključ (eng. *Primary key*) je ključ tipa entiteta koji može jedinstveno identificirati sva pojedinačna pojavljivanja entiteta u tipu entiteta.

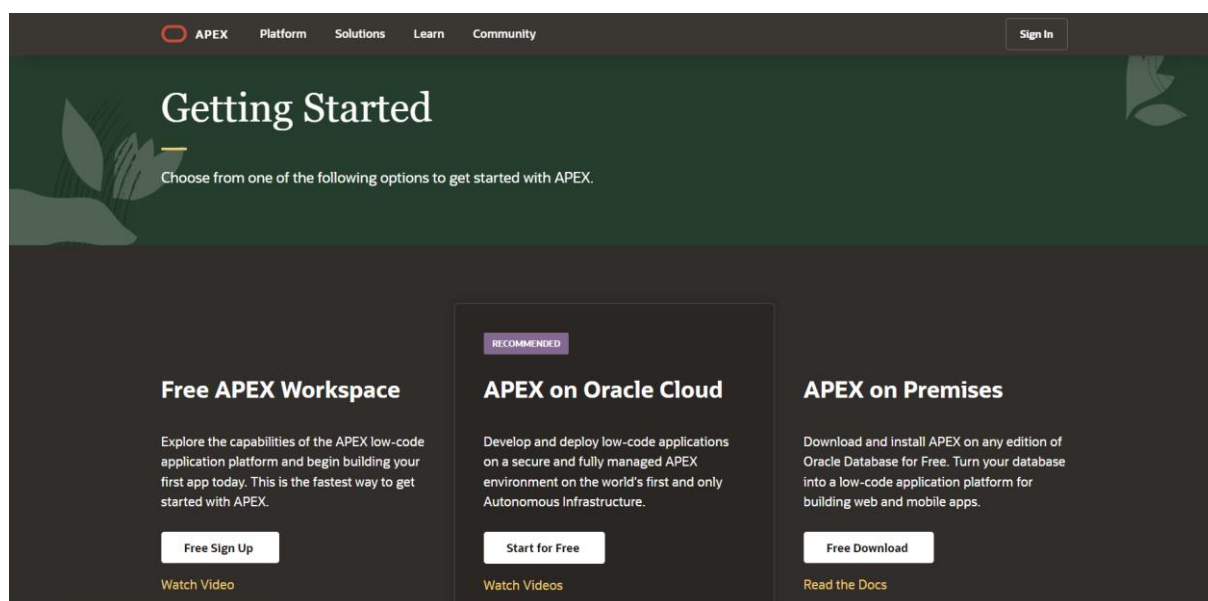
² Ako neki skup atributa u promatranoj relaciji nije ključ, ali je ključ u nekoj drugoj relaciji u modelu, onda se on naziva vanjski ključ (eng. *Foreign key*).

5. Izrada aplikacije

U ovom je poglavlju dan detaljan opis cjelokupnog procesa izrade poslovne aplikacije za rezervaciju smještaja, pri čemu su svi opisi popraćeni i snimkama zaslona koje ih vizualno predstavljaju.

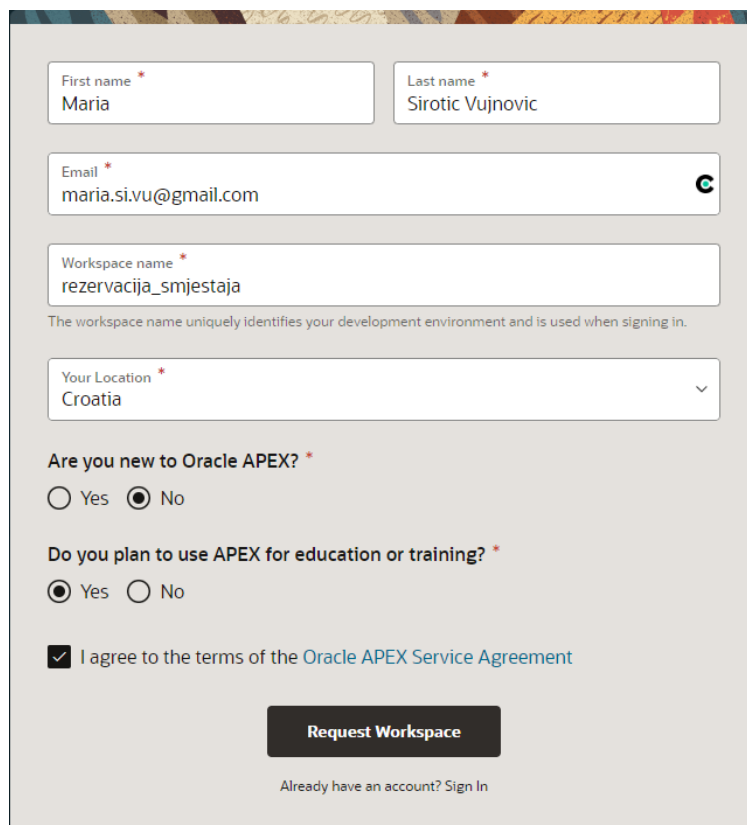
5.1. Postavljanje radnog okruženja

Alat Oracle APEX pruža višestruke načine upotrebe, a moguće ga je koristiti kao Oracle Cloud uslugu, putem internetskog preglednika ili lokalno putem preuzimanja na osobno računalo. Za potrebe izrade ovoga rada, pristup APEX alatu ostvaren je putem web preglednika. Na slici 6. prikazane su tri različite opcije za korištenje ovog alata, a iste su navedene na službenoj APEX web stranici (Oracle, 2023.).



Slika 6. Različiti načini korištenja Oracle APEX-a

Prvi korak u procesu izrade aplikacije obuhvaća postavljanje radnog okruženja (eng. *Workspace*), što uključuje podnošenje zahtjeva za novim radnim okruženjem. Novo radno okruženje možemo kreirati klikom na gumb *Free Sign Up* (unutra sekcije s naslovom *Free APEX Workspace*), prikazanim na slici 6. U okviru stranice koja nam se time otvara, potrebno je unijeti potrebne informacije – vlastito ime i prezime, e-mail adresu, odabrati naziv za buduće radno okruženje, te dodatno odabrati zemlju. Također pružamo informaciju o tome susrećemo li se s APEX alatom po prvi puta te namjeravamo li ga koristiti u svrhe obrazovanja ili ne. Nakon unosa ovih podataka na način prikazan slikom 7., klikom na gumb *Request workspace* završavamo s kreiranjem zahtjeva za novo radno okruženje.



First name *
Maria

Last name *
Sirotic Vujnovic

Email *
maria.si.vu@gmail.com

Workspace name *
rezervacija_smjestaja

The workspace name uniquely identifies your development environment and is used when signing in.

Your Location *
Croatia

Are you new to Oracle APEX? *
☐ Yes ☒ No

Do you plan to use APEX for education or training? *
☒ Yes ☐ No

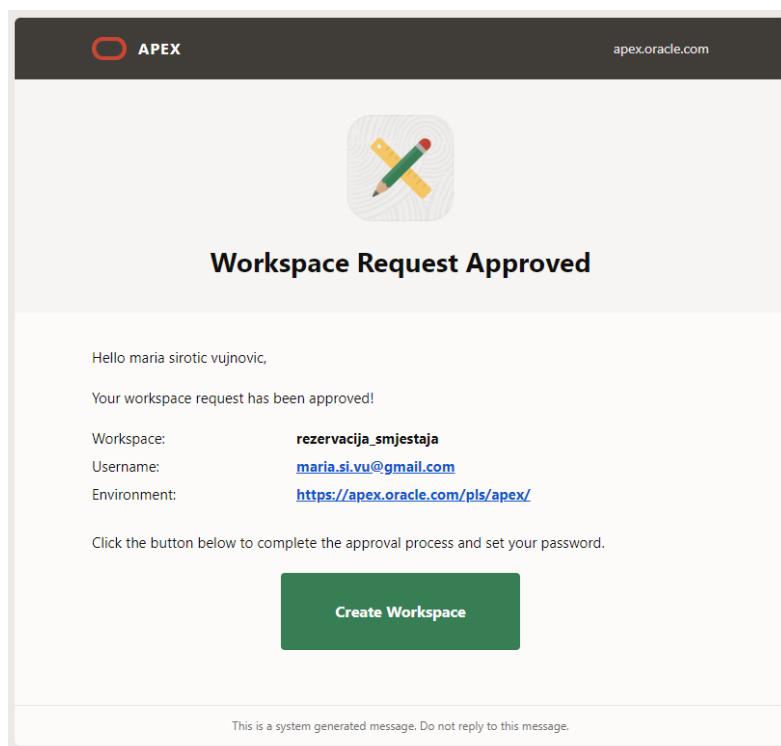
☒ I agree to the terms of the [Oracle APEX Service Agreement](#)

Request Workspace

Already have an account? [Sign In](#)

Slika 7. Slanje zahtjeva za novo radno okruženje

Nakon toga, na unesenu e-mail adresu stiže poruka (slika 8.) putem koje je moguće dovršiti proces stvaranja radnog okruženja. Klikom na gumb *Create Workspace* korisnik potvrđuje svoj zahtjev, nakon čega mu se prikazuje povratna poruka o uspješnom stvaranju novog okruženja.



APEX apex.oracle.com

Workspace Request Approved

Hello maria sirotic vujnovic,

Your workspace request has been approved!

Workspace: **rezervacija_smjestaja**
 Username: maria.si.vu@gmail.com
 Environment: <https://apex.oracle.com/pls/apex/>

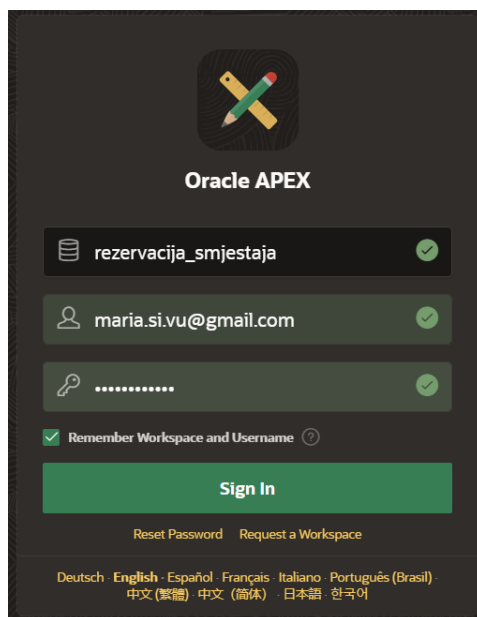
Click the button below to complete the approval process and set your password.

Create Workspace

This is a system generated message. Do not reply to this message.

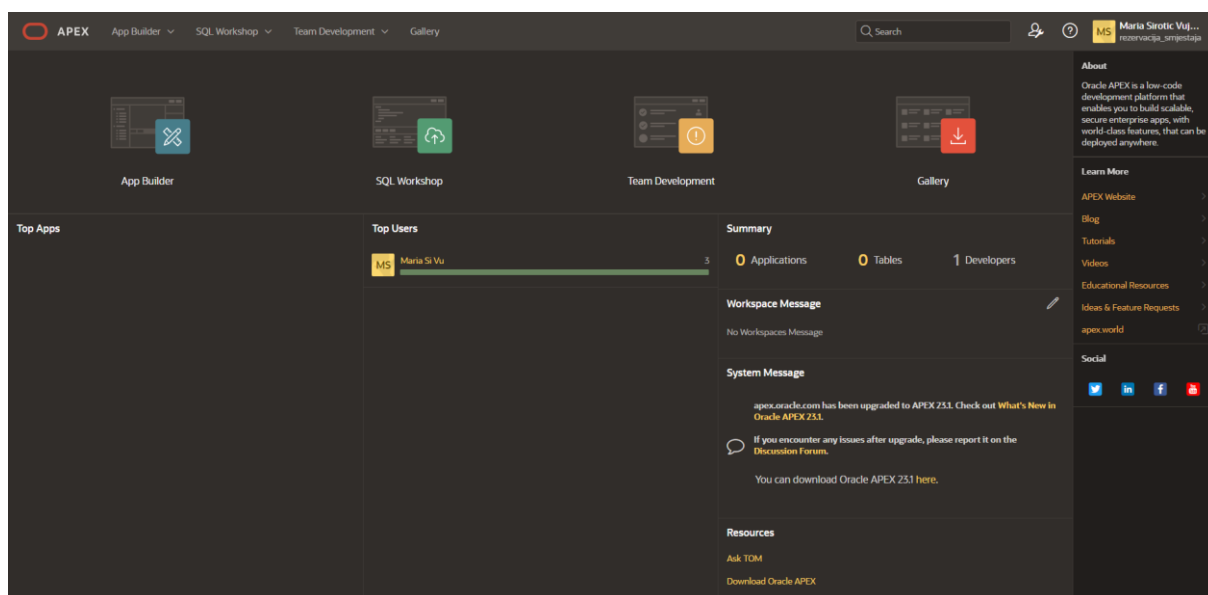
Slika 8. E-mail za završetak kreiranja radnog okruženja

Za prijavu unutar ovog radnog okruženja, potrebno je unijeti relevantne podatke u formu za prijavu u radno okruženje, kako je prikazano slikom 9. To uključuje naziv radnog okruženja, korisničko ime (e-mail adresu) i lozinku, respektivno.



Slika 9. Prijava u radno okruženje

Nakon unosa ovih podataka, korisnik će biti usmjeren na početni zaslon APEX radnog okruženja, prikazan slikom 10.



Slika 10. Početni zaslon radnog okruženja

Dva su ključna alata koja će biti korištena unutar ovog sučelja:

- *App Builder*
- *SQL Workshop*

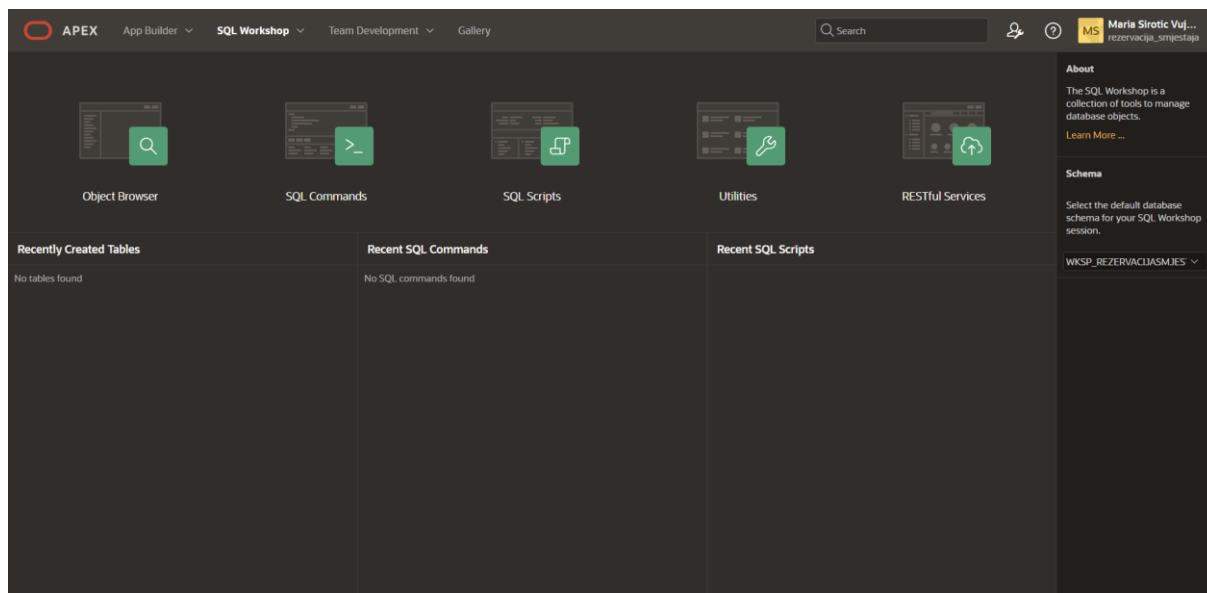
5.2. Kreiranje baze podataka – SQL workshop

Unutar alata SQL Workshop ističu se ovdje dva izuzetno važna alata:

- *Object Browser* – nudi mogućnost manipulacije bazom podataka putem grafičkog korisničkog sučelja
- *SQL Commands* – omogućuje izvršavanje SQL te PL/SQL blokova koda putem linijskog korisničkog sučelja

Oba navedena alata omogućuju pregled i manipulaciju bazom podataka, no koriste različite vrste sučelja. Object Browser alat pruža grafičko sučelje koje olakšava izvršavanje zadataka putem klikova i ispunjavanja formi. S druge strane, SQL Commands Tool pruža linijsko (tekstualno) sučelje za izvođenje zadataka putem ručnog unosa potrebnih SQL naredbi. Važno je napomenuti da SQL Commands Tool pruža napredne mogućnosti i omogućuje obavljanje složenijih zadataka. Međutim, njegova upotreba zahtijeva dublje razumijevanje SQL-a te same baze podataka s kojom se radi. Izbor jednog od ovih alata ovisi o vlastitim preferencijama, no moguće ih je koristiti u kombinaciji te tako iskoristiti njihove pune potencijale.

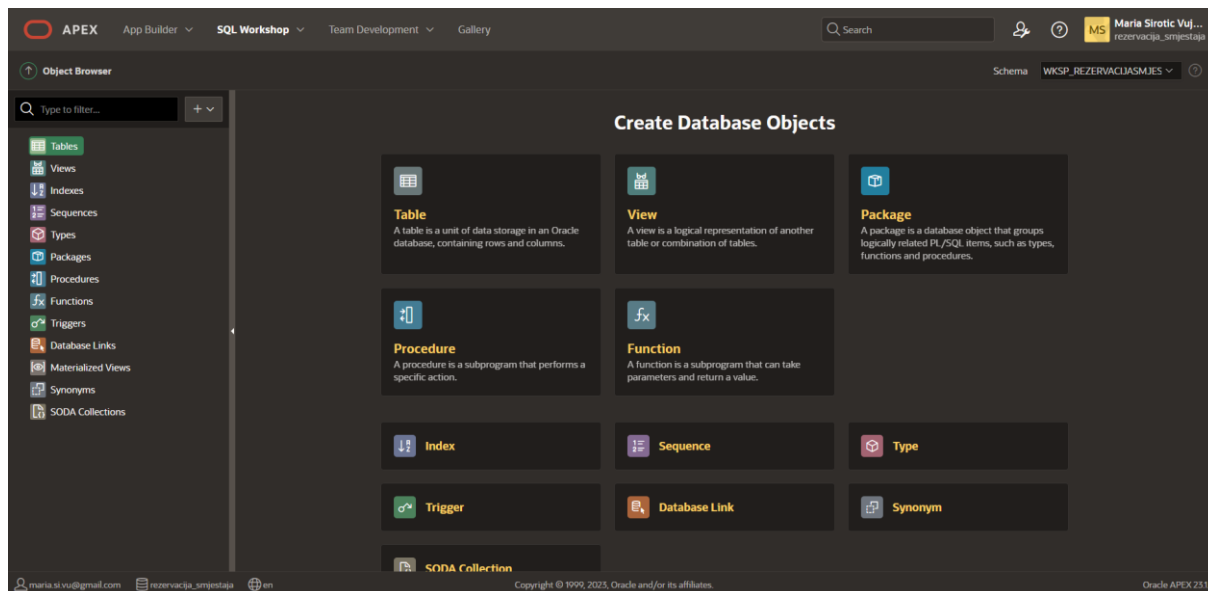
Da bismo pristupili SQL Workshop alatu, koristimo istoimeni naziv unutar glavne navigacije (vidljivo kao „SQL Workshop“ na slici 10.) ili kliknemo na veliki gumb SQL Workshop s početne stranice radnog okruženja. Klikom na ovu opciju, otvara se novi ekran prikazan slikom 11, koji nam omogućava pristup različitim alatima, uključujući alat *Object Browser* za pregledavanje i upravljanje bazom podataka te *SQL Commands* za izvođenje SQL upita izravno unutar sučelja.



Slika 11. Sučelje SQL Workshop alata

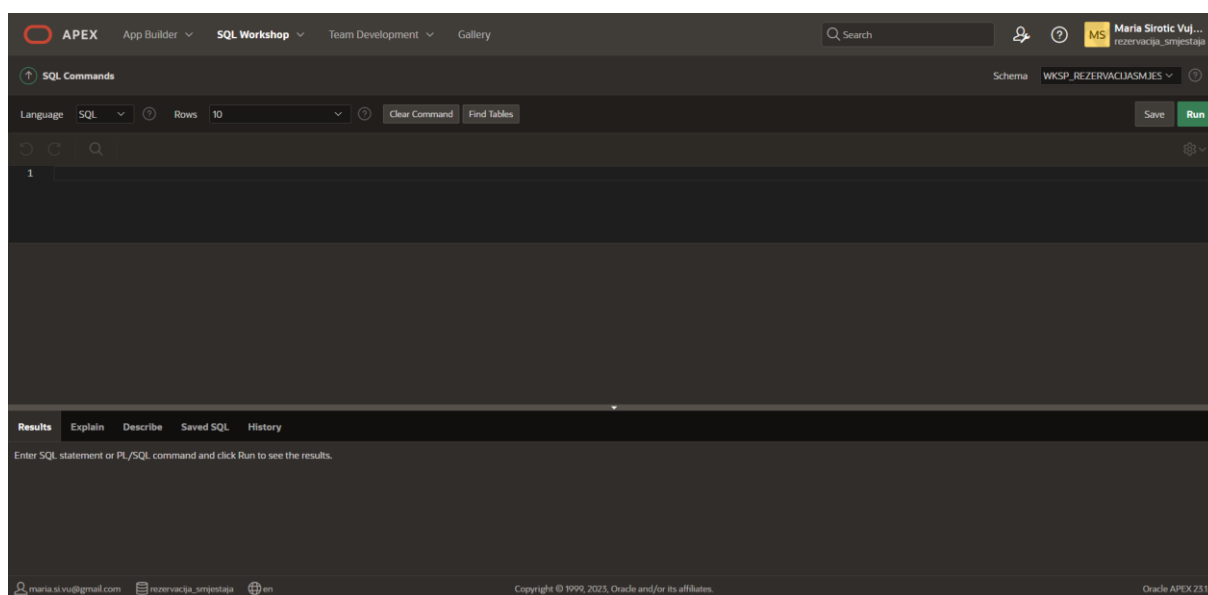
Object Browser alat pruža korisnicima mogućnost jednostavne, brze i efikasne interakcije s tablicama i drugim objektima unutar baze podataka. Do njega možemo doći klikom na gumb *Object Browser* (vidljiv na slici 11), a on nudi mnoge funkcionalnosti. Na primjer, za svaku tablicu pruža detaljne informacije o tipovima podataka i ograničenjima prisutnim u njezinim stupcima, nudi mogućnost unosa i manipulacije recima, indeksima i ulogama tablice. Osim

toga, alat omogućuje laku promjenu opisa ili sadržaja tablice. Također pruža funkcionalnost za dodavanje novih stupaca ili uklanjanje postojećih, kao i mogućnost ažuriranja istih. Sve ove akcije mogu se obaviti bez potrebe za kompleksnim SQL upitima ili dubokim tehničkim poznavanjem potrebnih tehnologija, čineći upravljanje bazom podataka intuitivnim i praktičnim. Sučelje ovog alata prikazano je kroz sliku 12.



Slika 12. Sučelje alata Object Browser

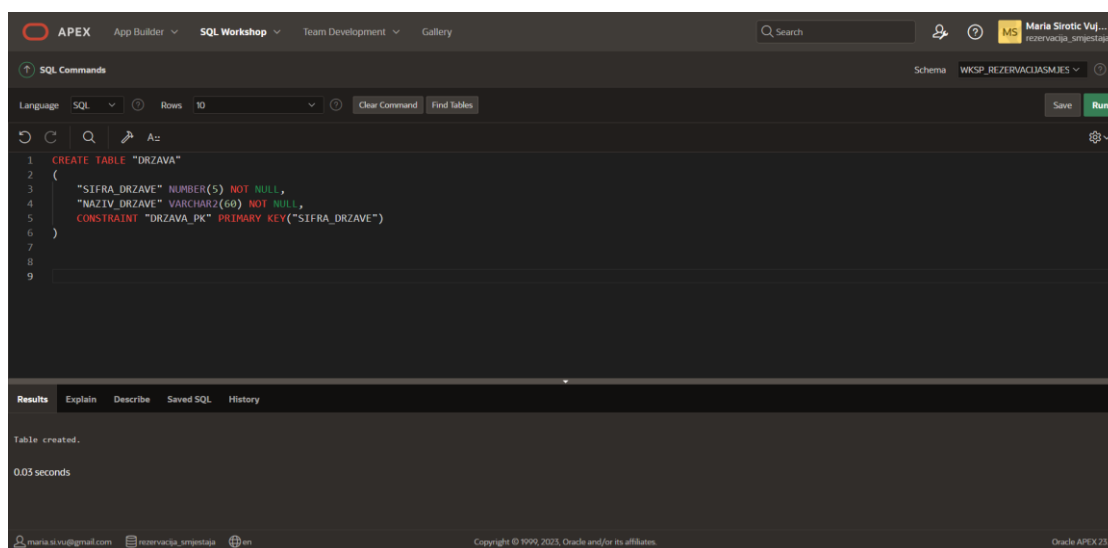
Drugi navedeni alat, *SQL Commands Tool*, pruža gotovo iste mogućnosti kao i prethodno opisani Object Browser alat, s obzirom na to da se sve radnje izvedene putem tog alata mogu prevesti u jednu ili više SQL izjava. Na vrhu ekrana, korisnici unose željeni SQL upit, čiji rezultati se potom prikazuju u donjem dijelu ekrana. Sučelje ovog alata prikazano je slikom 13. U suštini, SQL Command Tool omogućuje korisnicima da direktno izvršavaju SQL upite i manipuliraju podacima, dok istovremeno omogućuje pristup informacijama o strukturi baze podataka i njezinim objektima, čime olakšava administraciju i analizu podataka.



Slika 13. Sučelje alata SQL Commands

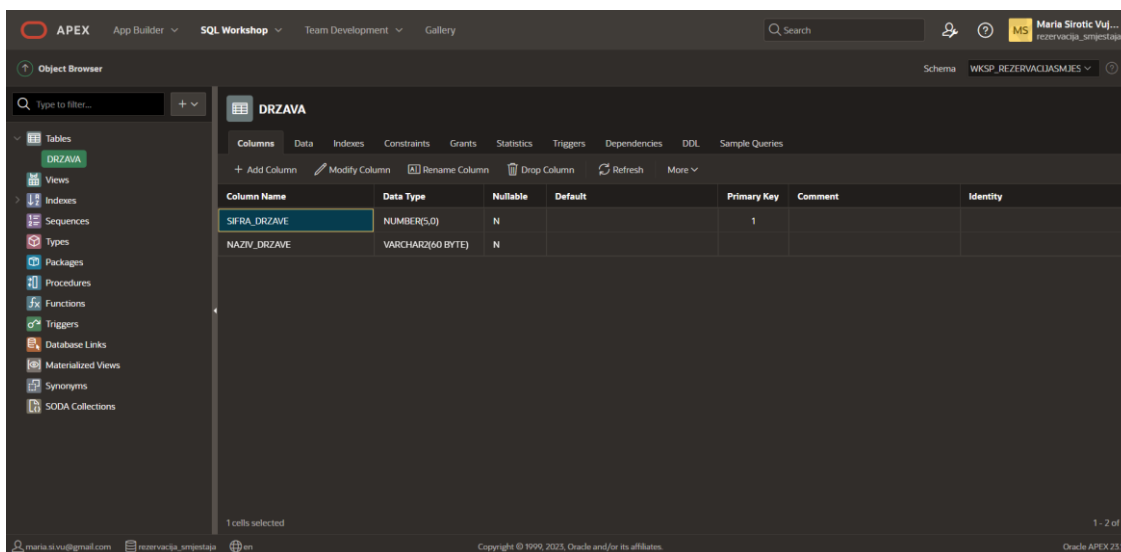
Moja preferencija je kreiranje tablica kroz alat SQL Browser. Ipak, radi boljeg razumijevanja, u nastavku ću također prikazati i opisati kako se isti zadatak može izvesti putem alata *SQL Commands*. Same tablice kreirane su sukladno ranije izrađenim modelima, posebno prateći prethodno dani relacijski model. Prilikom izrade tablica, važno je prvo stvoriti one najjednostavnije tablice, to jest one koje ne uključuju vanjske ključeve.

Slijedi primjer kako kreirati jednostavnu tablicu DRŽAVA putem navedenog alata. Postupak je trivijalan te uključuje nekoliko jednostavnih koraka. Najprije je potrebno unijeti odgovarajuću SQL naredbu (slika 14) koja definira strukturu tablice, uključujući njeno ime, definicije stupaca (njihove tipove podataka i duljinu) te potrebna ograničenja. Unesenu naredbu potrebno je potom izvršiti klikom na gumb *Run*, nakon čega alat pruža povratnu poruku o rezultatu, uključujući informaciju o uspješnom stvaranju tablice ili pak o eventualnim pogreškama koje je potom potrebno ispraviti te naredbu ponovno izvršiti.



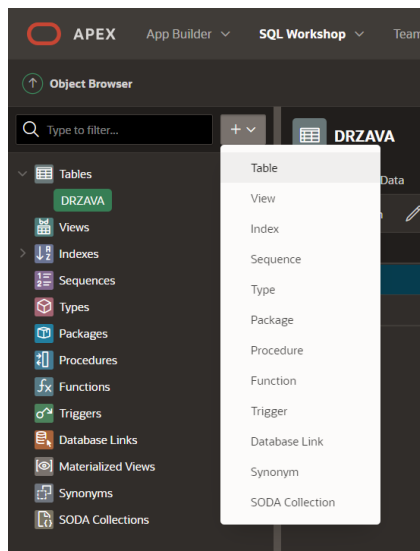
Slika 14. Kreiranje tablice DRŽAVA kroz SQL Commands alat

Unutar alata Object Browser sada možemo pregledati maločas kreiranu tablicu na način da iz panela s lijeve strane biramo *Tables* → *DRŽAVA*.



Slika 15. Prikaz tablice DRŽAVA u alatu Object Browser

Dalje, evo primjera kako kreirati tablicu kroz praktično grafičko sučelje alata Object Browser. Nakon što smo otvorili navedeni alat (*SQL Workshop* → *Object Browser*), iz panela s lijeve strane kliknemo na znak +. Ovo će otvoriti padajući izbornik iz kojega odaberemo opciju *Table*, s obzirom da želimo kreirati novu tablicu. Navedeni postupak prikazan je slikom 16. Potom je potrebno pratiti korake navedene u nastavku teksta.



Slika 16. Odabir opcije za kreiranje nove tablice

Odabirom opcije *Table* iz prethodnog koraka, otvara se novi dijaloški okvir, omogućavajući nam unos neophodnih podataka. Ovdje specificiramo naziv same tablice, imena stupaca i njihove tipove vrijednosti te duljinu. Također imamo opciju odabrati hoće li vrijednosti tog stupca biti dopuštene kao NULL vrijednosti. Osim toga, definiramo atribut koji će poslužiti kao primarni ključevi ove tablice – moguće je odabrati jedan ili više atributa kao primarne ključeve, ovisno o potrebama.

Primjerom sa slike 17. prikazano je kako korak po korak kreirati tablicu MJESTO. Počinjemo unosom naziva same tablice u polje *Table Name*, dok polja *Semantics* i *Comments* ostavljamo nepromijenjenima. Nakon toga slijede definicije stupaca unutar tablice. Primjerice, za atribut SIFRA_DRZAVE unosimo istoimeni naziv toga stupca u polje *Column Name*. Potom odabiremo tip vrijednosti kojeg će on sadržavati pod *Data Type* (u ovom slučaju *NUMBER*), a zatim definiramo i maksimalnu duljinu podataka koje će ovaj stupac pohranjivati. Za ovaj tip vrijednosti, polje *Precision* označava ukupan broj znamenki, a polje *Scale/Length* definira broj decimalnih mjesta. Kod tipa vrijednosti *VARCHAR2*, duljinu podataka definiramo unutar polja *Scale/Length*. Preostalo je još označiti da ovaj stupac ne smije poprimati NULL vrijednosti te da je on dio primarnog ključa ove tablice, a što učinimo označivanjem potvrdnih okvira za polja *Not Null* i *Primary Key*. Na ekvivalentan način definiramo i preostala dva stupca, od kojih je POSTANSKI_BROJ također dio primarnog ključa tablice, dok se za NAZIV_DRZAVE unosi niz znakova koji predstavlja sam naziv države pa je ovaj stupac tipa podataka *VARCHAR2*. Klikom na gumb *Create Table* završavamo s postupkom kreiranja tablice te je istu sada moguće pronaći u panelu s lijeve strane, u popisu svih tablica – *Tables*.

Create Table

Table Name

MJESTO

Semantics

CHAR

Comments

Columns

Edit

Add Column

	Column Name	Data Type	Precision	Scale / Length	Not Null	Primary Key	Default Expression	Comments
	SIFRA_DRZAVE	NUMBER	5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	POSTANSKI_BROJ	NUMBER	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	NAZIV_MJESTA	VARCHAR2		60	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

1 rows selected

Total 3

Double-click on any cell or click on the Edit button to enable editing.

Cancel

Preview SQL

Create Table

Slika 17. Kreiranje tablice MJESTO kroz alat Object Browser

Tablica MJESTO predstavlja slab tip entiteta ovisan o tablici DRZAVA. Drugim riječima, primarni ključ tablice DRZAVA dio je primanog ključa tablice MJESTO. Kako bi se uspješno povezale, omogućavajući da putem tablice MJESTO pristupimo atributu SIFRA_DRZAVE iz tablice DRZAVA, potrebno je definirati vanjski ključ unutar tablice MJESTO. Da bismo postigli ovu vezu, koristimo ograničenje vanjskog ključa. Klikom na karticu *Constraints* unutar alata (otvorimo tablicu MJESTO→*Constraints*), možemo dodati ovo potrebno ograničenje. Ovdje također imamo mogućnost pregleda svih već postojećih ograničenja, što olakšava praćenje i upravljanje cjelokupnom strukturom baze podataka.

Kartica *Constraints* nudi mogućnosti pregledavanja, dodavanja i uklanjanja ograničenja vezanih uz pojedinu tablicu. Klikom na gumb *Create* započinjemo s procesom dodavanja novog ograničenja. Prilikom dodavanja ograničenja najprije je potrebno definirati njegovo jedinstveno ime u polju *Name*. Osobno sam koristila sljedeći princip nazivanja vanjskih ključeva: *NAZIVATRIBUTA_NAZIVTABLICE_FK*. Primjerice, ukoliko je atribut SIFRA_DRZAVE vanjski ključ u tablici MJESTO, naziv takvog ograničenja glasi SDRZAVE_MJESTO_FK. Nakon toga, u polju *Constraint Type* odabiremo iz padajućeg izbornika opciju *Foreign Key*, budući da dodajemo ograničenje vanjskoga ključa. Zatim u polju *Foreign Key Column(s)* definiramo stupac (ili stupce) nad kojim želimo postaviti ograničenje vanjskoga ključa. U ovom slučaju, odabiremo samo stupac SIFRA_DRZAVE. Pomoću okvira s lijeve strane u kojem su navedeni svi atributi tablice MJESTO, mišem odabiremo one na koje želimo primijeniti ograničenje te ih klikom na znak > prebacujemo u desni okvir. U polju *Reference Table Name* (klikom na isto otvara se padajući popis svih tablica koje postoje u trenutnoj bazi podataka)

odabiremo tablicu iz koje želimo dohvaćati podatke – u ovom je slučaju to tablica DRZAVA. Zatim, unutar posljednjeg polja, iz tablice DRZAVA odabiremo atribut putem kojega će se ostvariti povezivanje. Taj se atribut u tablici DRZAVA također naziva SIFRA_DRZAVE. Slika 18 prikazuje konačnu ispunjenu formu za dodavanje novog ograničenja vanjskog ključa u tablicu MJESTO. Klikom na gumb *Apply* uspješno primjenjujemo definirano ograničenje.

Create Constraint

Owner: WKSP_REZERVACIJASMJESTAJA

Table: MJESTO

Name: SDRZAVE_MJESTO_FK

Constraint Type: Foreign Key

Delete Behavior: ☒ Disallow Delete ☐ Cascade Delete ☐ Set Null on Delete

Foreign Key Column(s): POSTANSKI_BROJ, NAZIV_MJESTA

Reference Table Name: DRZAVA

Reference Table Column(s): NAZIV_DRZAVE, SIFRA_DRZAVE

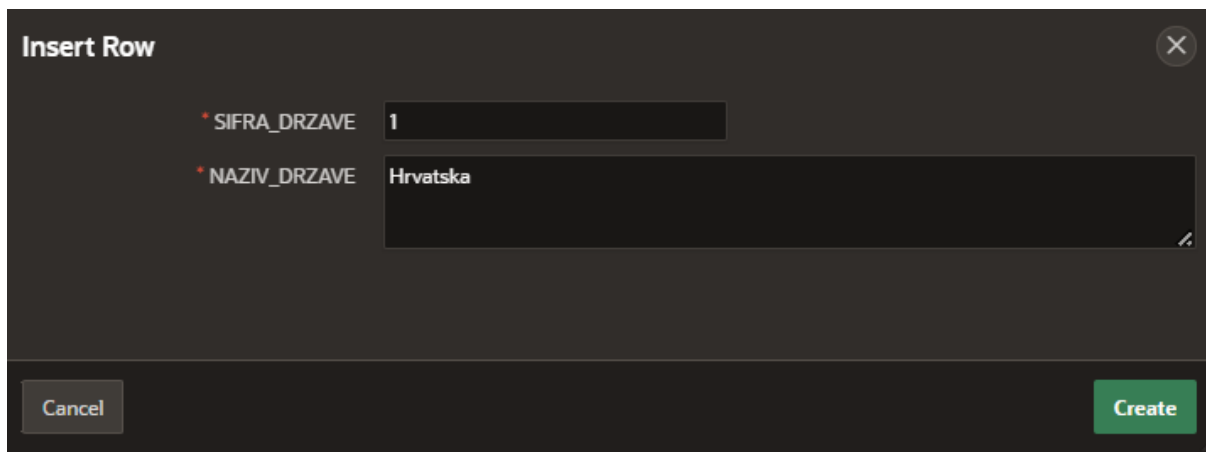
Buttons: Cancel, Apply

Slika 18. Čarobnjak za dodavanje ograničenja vanjskog ključa u tablicu MJESTO

Kako bismo se uvjerali da je ograničenje ispravno postavljeno, unesimo testne podatke za verifikaciju. Ako je ograničenje uspješno postavljeno, prilikom unosa vrijednosti u stupac SIFRA_DRZAVE u tablici MJESTO, nužno je unijeti jednu od vrijednosti koja već postoji u tablici DRZAVA, točnije zapisana je unutar njezinog stupca SIFRA_DRZAVE.

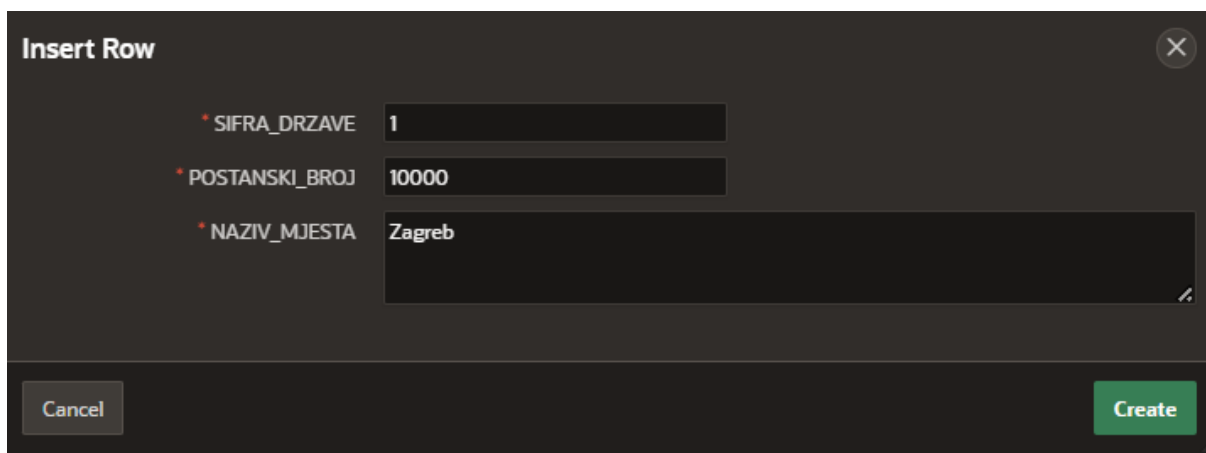
Započnimo unosom nove države u tablicu DRZAVA na sljedeći način. Kliknemo na tablicu DRZAVA, zatim odaberemo karticu *Data*, te nakon toga kliknemo na gumb + *Insert Row*.

Unutar otvorenog okvira unosimo tražene podatke prema slikovnom prikazu u nastavku te kliknemo na gumb *Create*. U ovom primjeru, unesena je država sa šifrom 1 i nazivom *Hrvatska*.

The image shows a dark-themed 'Insert Row' dialog box. It has a title bar with a close button (X) on the right. Inside, there are two labeled input fields: '* SIFRA_DRZAVE' with the value '1' and '* NAZIV_DRZAVE' with the value 'Hrvatska'. At the bottom, there are two buttons: 'Cancel' on the left and 'Create' on the right.

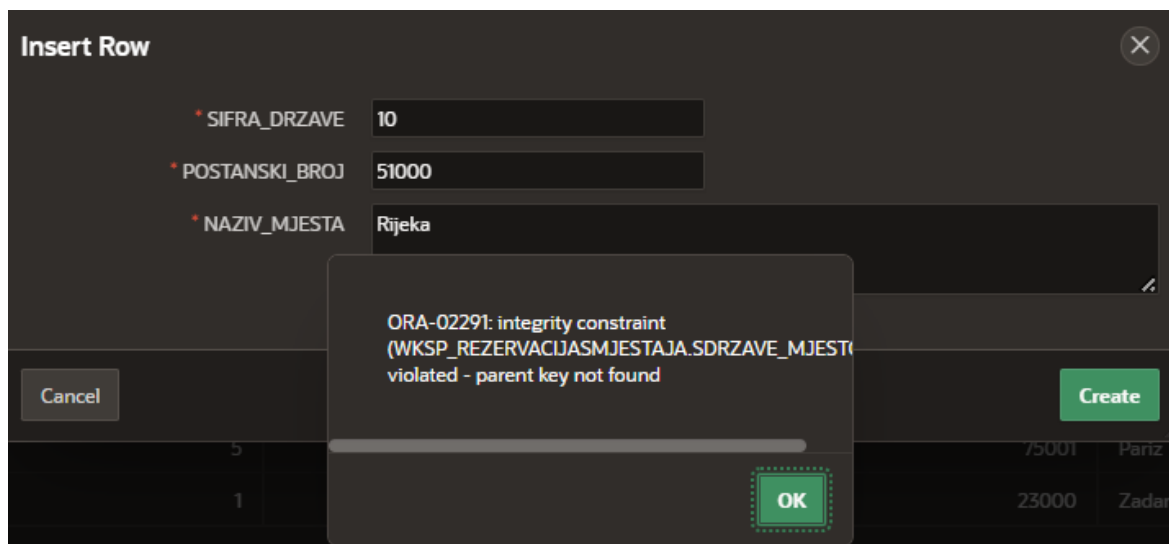
Slika 19. Unos nove države u tablicu DRZAVA

Sada pokušajmo dodati novo mjesto u istoimenu tablicu. Na primjer, unesimo grad Zagreb koji se nalazi u Hrvatskoj. Za šifru države unesimo vrijednost 1, što je ista vrijednost koja je pridružena zemlji *Hrvatska* u tablici DRZAVA. Klikom na gumb *Create* dodajemo novi zapis ovoj tablici, a u što se možemo uvjeriti i putem kartice *Data* koja prikazuje sve podatke zapisane unutar ove tablice.

The image shows a dark-themed 'Insert Row' dialog box. It has a title bar with a close button (X) on the right. Inside, there are three labeled input fields: '* SIFRA_DRZAVE' with the value '1', '* POSTANSKI_BROJ' with the value '10000', and '* NAZIV_MJESTA' with the value 'Zagreb'. At the bottom, there are two buttons: 'Cancel' on the left and 'Create' on the right.

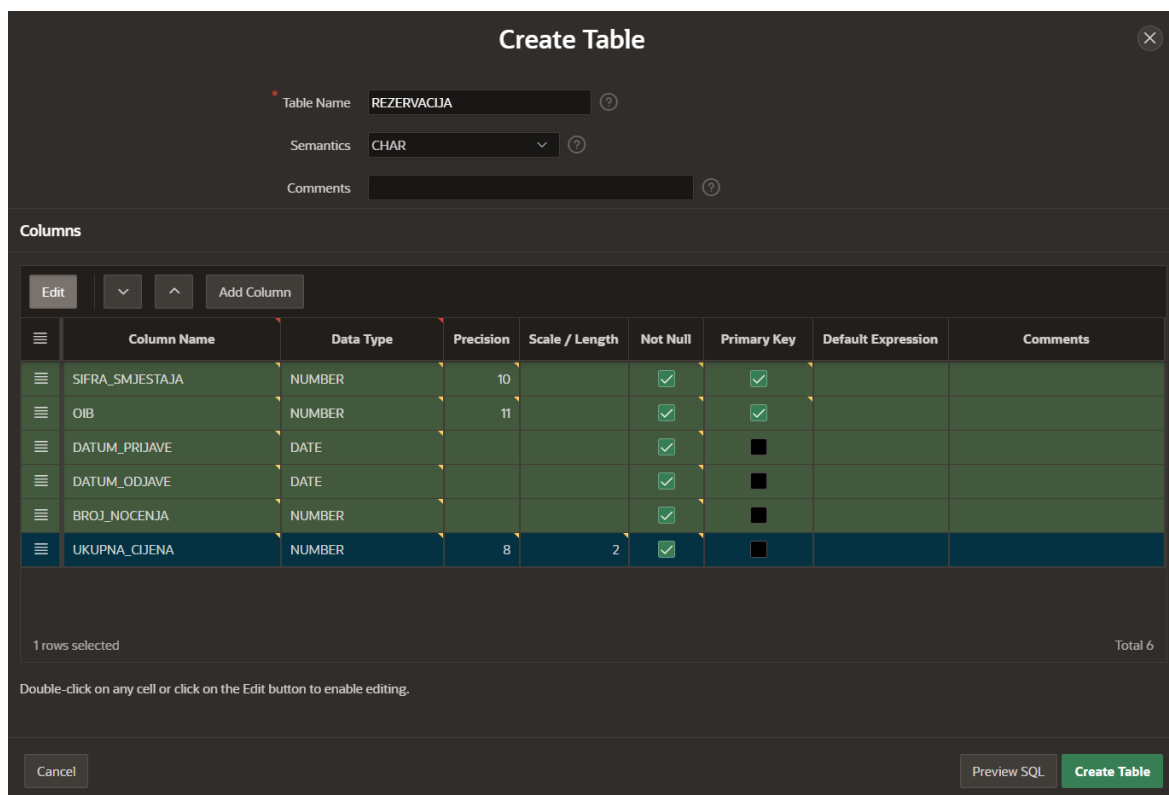
Slika 20. Unos novog mjesta u tablicu MJESTO

Pokušajmo sada dodati novo mjesto, ali ovog puta unesimo u polje SIFRA_DRZAVE vrijednost koja ne postoji unutar istoimenog stupca u tablici DRZAVA. Na primjer, unesimo vrijednost 4 (dosada u tablici DRZAVA postoji samo zapis o jednoj zemlji sa šifrom 1). Klikom na *Create*, pojavit će se poruka o grešci, prikazana slikom 21, koja korisnika obavještava da roditeljski ključ nije pronađen te da je narušeno ograničenje definiranog vanjskog ključa.



Slika 21. Pogreška prilikom narušavanja ograničenja vanjskog ključa

Na ekvivalentan način kreiramo i preostale tablice, a slijedi još primjer kreiranja agregacija REZERVACIJA, s obzirom da se agregacije smatraju složenim tipovima entiteta. Kao i do sada, postupak započinjemo odabirom opcije za kreiranje nove tablice. Unosimo sve potrebne stupce koje će ova tablica sadržavati te definiramo njihova svojstva, kako je prikazano slikom 22. Kao primarni ključ ove tablice odabiremo dva stupca, s obzirom da agregacija posjeduje složeni primarni ključ, u ovome slučaju sastavljen od atributa SIFRA_SMJESTAJA te OIB.



Slika 22. Ispunjena forma za kreiranje tablice REZERVACIJA

Sada je potrebno osvrnuti se na pripadna ograničenja vanjskoga ključa, a koja je potrebno definirati kako bismo osigurali preuzimanje vrijednosti stupaca SIFRA_SMJESTAJA i OIB iz

njima pripadnih tablica. Slika 23. prikazuje postupak dodavanja ograničenja vanjskog ključa na atribut SIFRA_SMJESTAJA. Na isti način dodajemo i ograničenje za stupac OIB, samo ovoga puta pod referentom tablicom odabiremo tablicu KORISNIK te njezin atribut OIB pod poljem Reference Table Column(s).

Create Constraint

Owner: WKSP_REZERVACIJASMJESTAJA

Table: REZERVACIJA

Name: SSMJESTAJ_REZERVACIJA_FK

Constraint Type: Foreign Key

Delete Behavior:

- ☐ Disallow Delete
- ☒ Cascade Delete
- ☐ Set Null on Delete

Foreign Key Column(s):

OIB	←	SIFRA_SMJESTAJA
DATUM_PRIJAVE	>>	
DATUM_ODJAVE	>	
BROJ_NOCENJA	<	
UKUPNA_CIJENA	<<	

Reference Table Name: SMJESTAJ

Reference Table Column(s):

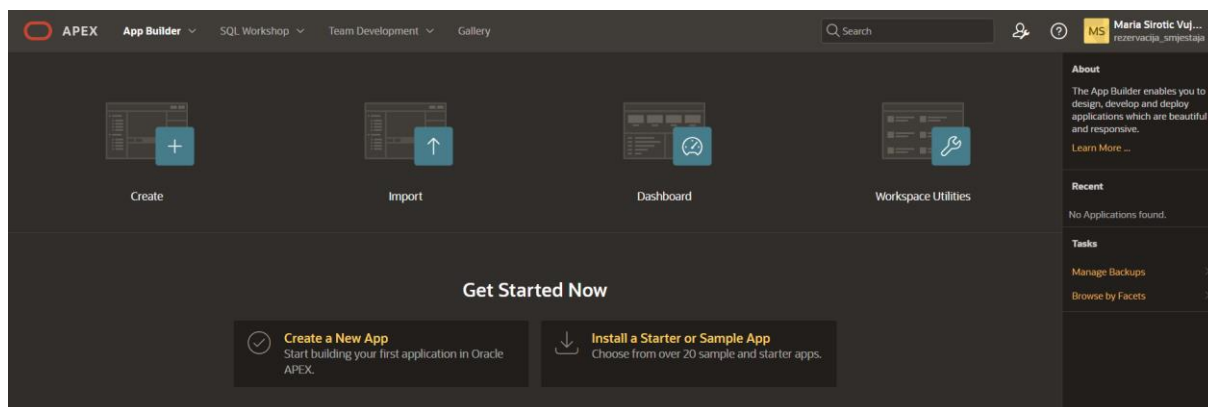
NAZIV_SMJESTAJA	←	SIFRA_SMJESTAJA
OPIS_SMJESTAJA	>>	
ADRESA_SMJESTAJA	>	
TIP_SOB	<	
MAX_BROJ_GOSTIJU	<<	
SIFRA_TIPA_SMJESTAJA		
SIFRA_DRZAVE		
POSTANSKI_BROJ		

Buttons: Cancel, Apply

Slika 23. Kreiranje ograničenja vanjskog ključa za stupac SIFRA_SMJESTAJA u tablici REZERVACIJA

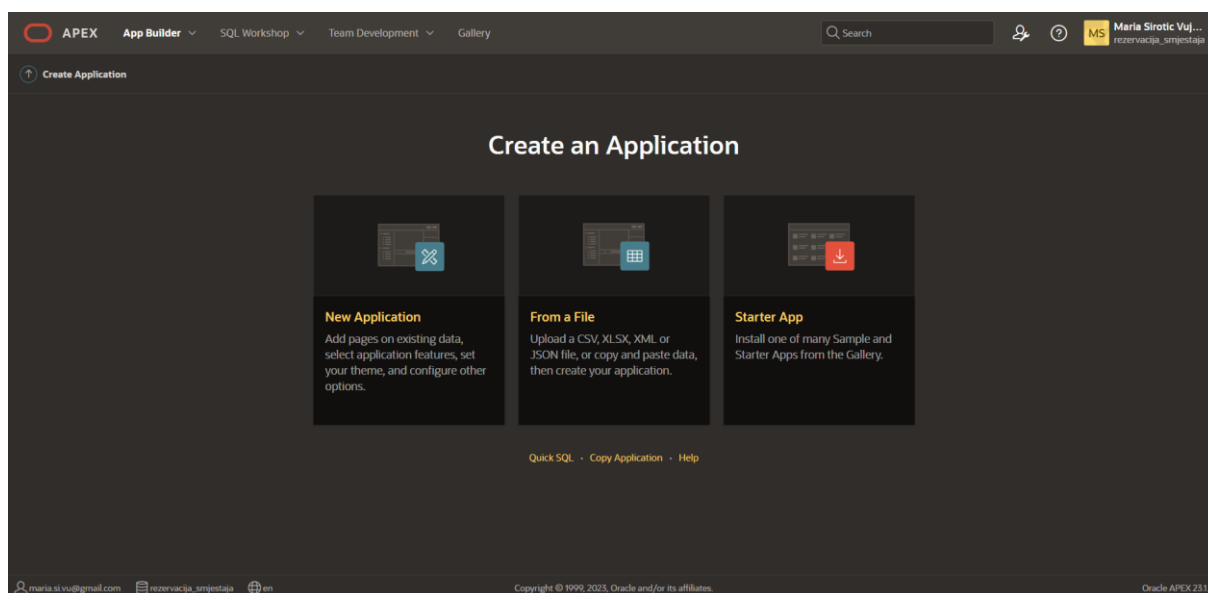
5.3. Kreiranje aplikacije

Nakon što je kreirana baza podataka, možemo započeti s izradom same aplikacije nad njom. Čarobnjaci kojima stvaramo aplikacije i stranice nazivaju se *Create Application* te *Create Page*, a dio su alata *App Builder*. Odabirom ovog alata iz glavne navigacijske trake (ili klikom na istoimeni gumb s početnog ekrana radnog okruženja), otvara se njegovo sučelje, prikazano slikom 24. Unutar toga alata sada možemo pristupiti gumbu *Create* čime se otvara čarobnjak *Create Application*, putem kojega stvaramo novu aplikaciju.



Slika 24. Sučelje alata App Builder

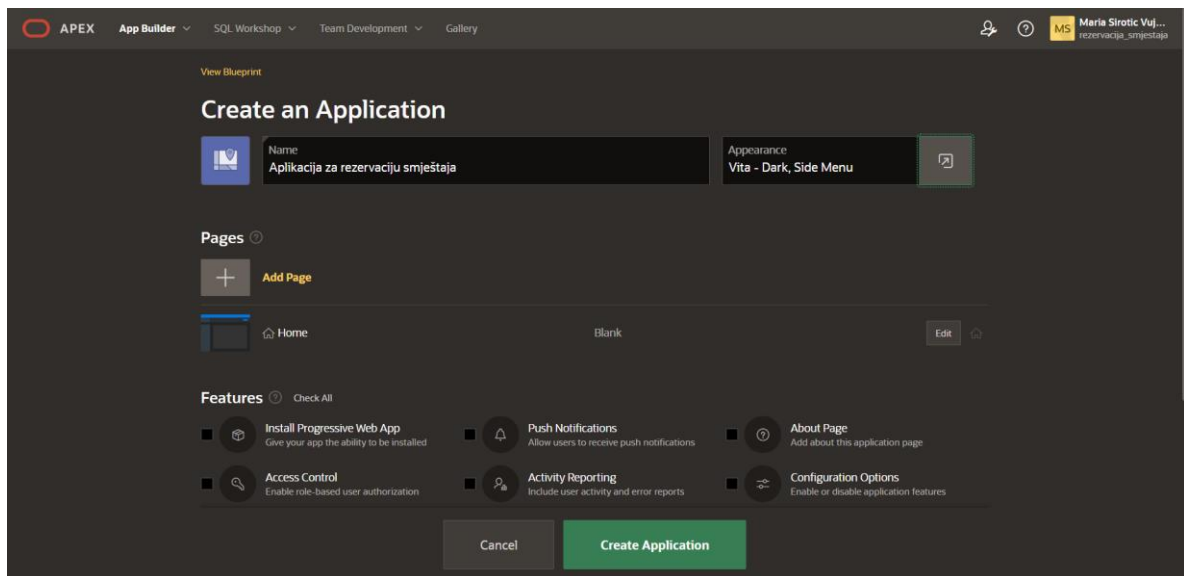
Čarobnjak od nas najprije zahtijeva da se odlučimo za vrstu aplikacije koju želimo kreirati, birajući između jedne od ponuđenih triju opcija. Ovdje je potrebo odabrati prvu od njih – opciju *New Application*, s obzirom da želimo kreirati novu aplikaciju i to od nule. Ostale opcije za kreiranje aplikacije unutar alata APEX prikazane su putem slikovnog prikaza 25.



Slika 25. Mogućnosti prilikom kreiranja nove aplikacije

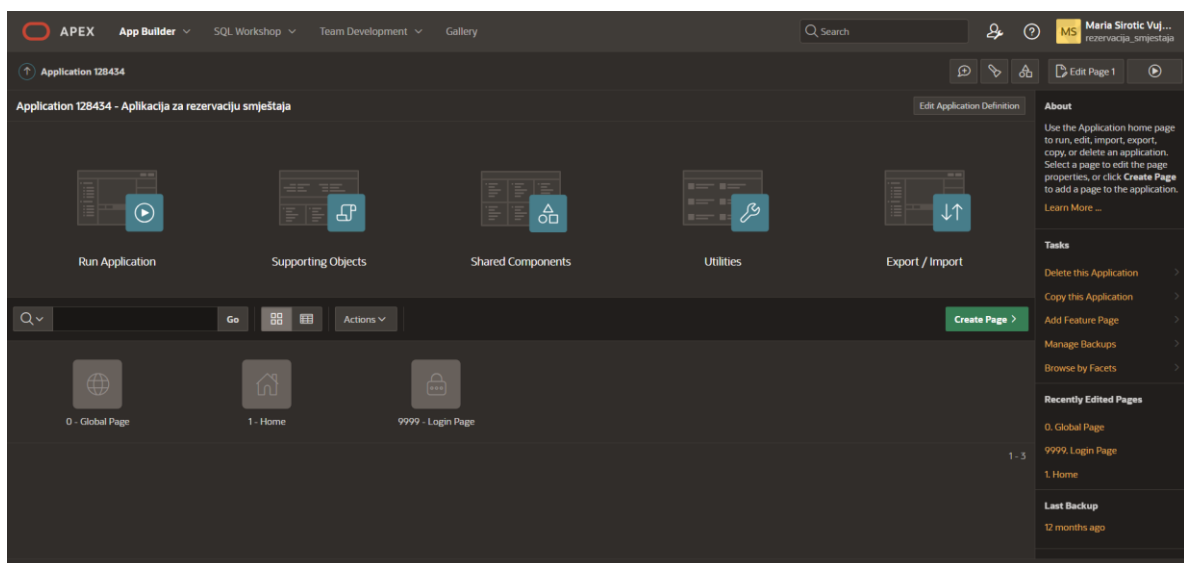
Unutar sljedećeg zaslona potrebno je definirati određena svojstva aplikacije. U prvo polje unosimo željeno ime aplikacije – ovdje je ono navedeno kao *Aplikacija za rezervaciju smještaja*. Također, moguće je odlučiti se za boju i izgled ikone aplikacije. Nadalje, pod *Appearance* definiramo koju će temu te vrstu navigacije aplikacija koristiti, pri čemu sam se ja ovdje odlučila za *Vita – Dark* temu te bočnu navigaciju u obliku panela s lijeve strane.

Unutar *Pages* sekcije ne vršimo nikakve promjene. Ovdje je već po zadanim postavkama definirano da će se aplikacija za početak sastojati od jedne prazne stranice, nazvane *Home*, što implicira da će ona služiti kao početna stranica. Stranice možemo dodavati naknadno, prema potrebi, pa je ovu sekciju trenutno najbolje ostaviti onakvom kakva je. Nadalje, sve postavke pod sekcijom *Features* također je moguće naknadno izmijeniti te nema potrebe specificirati ih u ovom trenutku, a isto vrijedi i za opciju *Settings*. Preostalo je završiti ovaj proces klikom na gumb *Create Application*. Slika 26. prikazuje maloprije opisani postupak.



Slika 26. Definiranje postavki nove aplikacije

Ukoliko se sada vratimo na sučelje alata *App Builder*, možemo se uvjeriti kako putem njega sada možemo pristupiti maloprije kreiranoj aplikaciji. Klikom na tu aplikaciju (iz popisa svih aplikacija), otvara nam početno sučelje za istu, prikazano slikom 27, putem kojega možemo pristupiti stranicama aplikacije. U ovom trenutku, moguće je pristupiti trima stranicama: globalnoj stranici, početnoj stranici te stranici za prijavu, prikazanima putem odgovarajućih ikona. Također, putem ovog sučelja možemo ažurirati generalne postavke aplikacije, klikom na gumb *Edit Application Definition*. Klikom na gumb *Run Application* pokrećemo samu aplikaciju. Ulaskom na svaku pojedinu stranicu također dolazimo do gumba *Run*, kojim je moguće pokretati pojedine stranice aplikacije. Prilikom prvog klika na gumb na *Run* potrebno je unijeti podatke za prijavu kako bismo pristupili aplikaciji, s obzirom da smo prilikom kreiranja aplikacije pod poljem *Authorization* ostavili definirano *Authorization by Application Express Accounts*, što znači da aplikaciju mogu pokretati samo one osobe koje imaju kreiran korisnički račun unutar radnog okruženja. Autorizacija korisnika biti će implementirana i opisana kroz kasnija poglavlja.



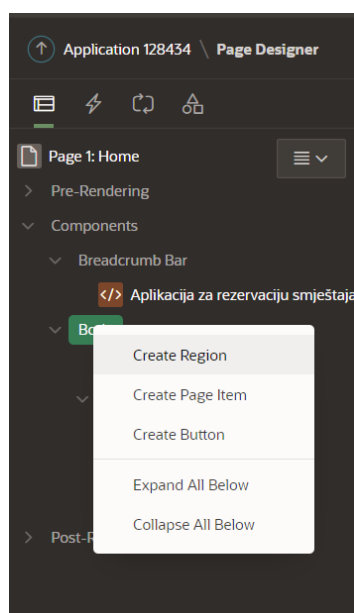
Slika 27. Početno sučelje aplikacije

Ukoliko aplikaciji želimo pak pristupiti kao krajnji korisnik, to činimo putem URL adrese. Svakoj aplikaciji pridružen je jedinstveni ID, a i svakoj njezinoj stranici također je pridružena jedinstvena identifikacijska oznaka, odnosno njoj svojstveni ID. Tri trenutne stranice ove aplikacije imaju ID vrijednosti 0 (Global Page), 1 (Home Page) i 9999 (Login Page). Kako bismo aplikaciji pristupili putem URL-a, potrebno je u samu adresu uvrstiti ID same aplikacije i ID stranice kojoj pristupamo, odvojenih dvotočkom. Prema tome, URL za pristup početnoj stranici ove aplikacije glasio bi: <https://apex.oracle.com/pls/apex/f?p=128434:1>

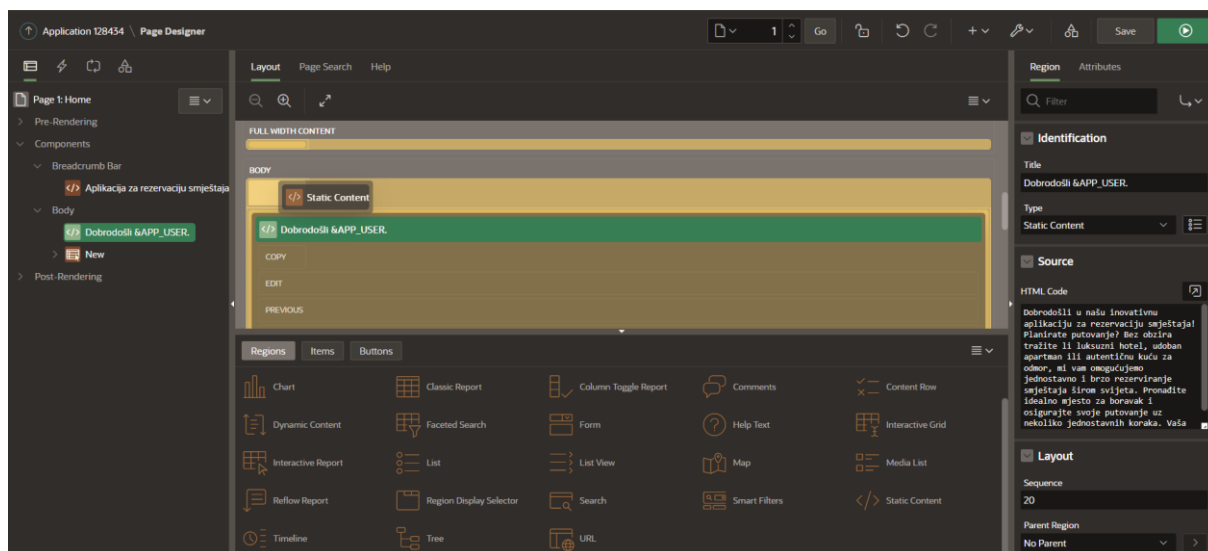
5.4. Regije statičnog sadržaja i klasična izvješća

Prilikom izrade aplikacije, najprije je uređena početna stranica koja već postoji unutar aplikacije (stvorena je prilikom kreiranja aplikacije). Unutar nje dodana je nova regija koja će prikazivati statični sadržaj, odnosno dodana je nova *Static Content* regija. Regije koristimo kako bismo putem njih prikazivali određene sadržaje unutar aplikacije. Najprije ćemo upoznati regija tipa *Static Content* koje prikazuju jednostavan tekst ili pak formatirani HTML kod.

Jedan od načina dodavanja regija na stranicu jest putem panela s lijeve strane. Primjerice, ukoliko regiju želimo dodati unutar sekcije *Body*, desnim klikom miša odaberemo tu regiju te potom iz padajućeg izbornika odabiremo opciju *Create Region*, (vidi sliku 28). Drugi je način dodavanja regije putem opcije *drag and drop* na način da iz galerije svih moguća tipova regija (smještene na dnu ekrana) odaberemo onu koju želimo dodati te je povučemo na željeno mjesto unutar stranice, odnosno *Layout* panela (vidi sliku 29). Kao naslov ove regije postavljena je vrijednost *Dobrodošli &APP_USER*, kako bismo najavili dobrodošlicu korisniku. Varijabla *&APP_USER* sadrži korisničko ime trenutno prijavljenog korisnika aplikacije. Također, u panelu s desne strane, pod poljem *Source* → *HTML Code* unesen je kratak tekst koji će biti prikazan unutar ove regije, a što možemo vidjeti na slici 29.



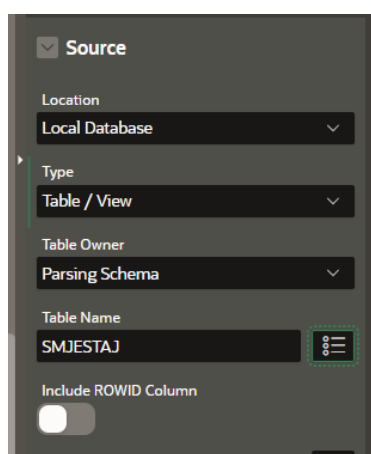
Slika 28. Dodavanje nove regije



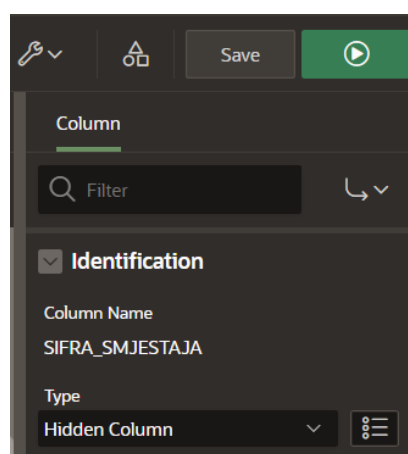
Slika 29. Dodavanje nove regije putem drag and drop opcije

Potom je na početnu stranicu dodan i prikaz smještaja, kako bi korisnik dobio informaciju koje su to smještajne jedinice koje je moguće rezervirati. Ovo je definirano kao statično izvješće, pri čemu je korisniku omogućeno filtriranje dostupnih smještajnih jedinica po mjestu ili pak tipu smještajnih jedinica. Najprije je dodana statična regija, kojoj je definiran samo naziv → *Filtriranje smještaja*. Zatim je unutar nje kreirana i *Classic Report* regija, na način da je ista povučena iz popisa svih regija (*Regions*) s dna ekrana. *Classic Report* regije koristimo za prikazivanje statičnih izvješća, što znači da se putem njih korisniku prikazuju potrebni podaci, ali mu se ne nude opcije dodavanja, uređivanja ili brisanja sadržaja.

Dodano statično izvješće prikazivat će podatke o smještajnim jedinicama pa u panelu s desne strane pod poljem *Identification* → *Type* za sada možemo ostaviti *Table* te odabrati tablicu SMJESTAJ (slika 30). Stupac SIFRA_SMJESTAJA definiran je kao *Hidden Column* (kako se on ne bi prikazivao krajnjem korisniku), dok je ostalim stupcima izmijenjen njihov naziv kako bi se aplikacija učinila pristupačnijom krajnjem korisniku. Postavljanje stupca kao skrivenog činimo putem panela s desne strane zaslona, pri čemu je potrebno odabrati željeni stupac te zatim pod sekcijom *Identification* → *Type* odabrati opciju *Hidden Column* (slika 31.)



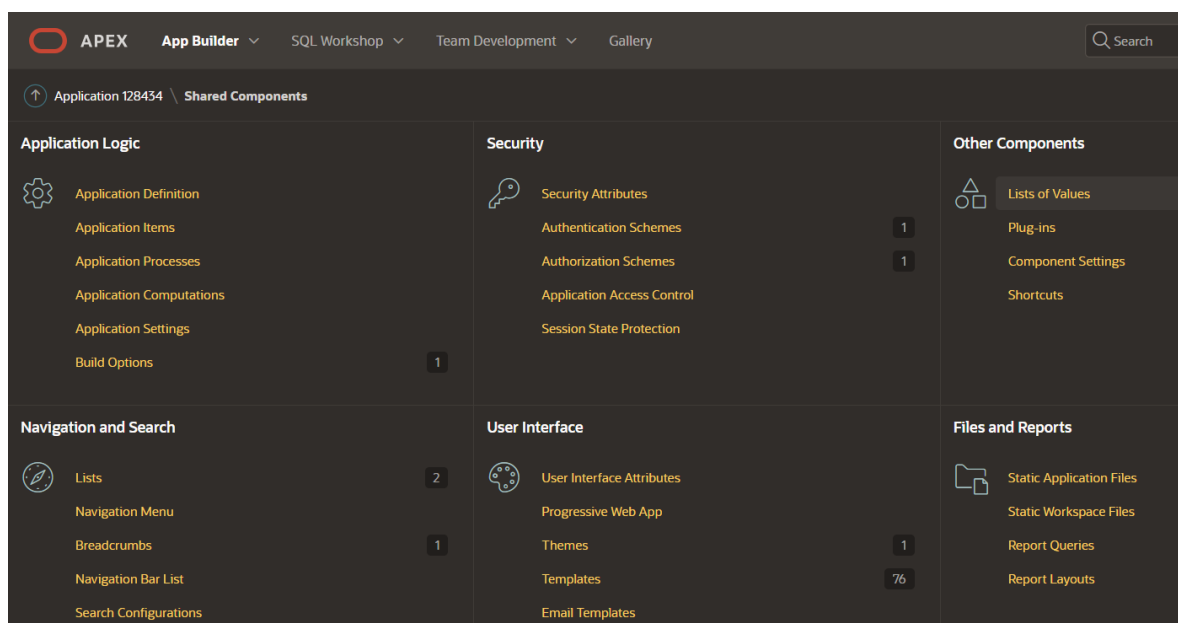
Slika 30. Definiranje izvora podataka za statično izvješće



Slika 31. Postavljanje stupca kao skrivenog

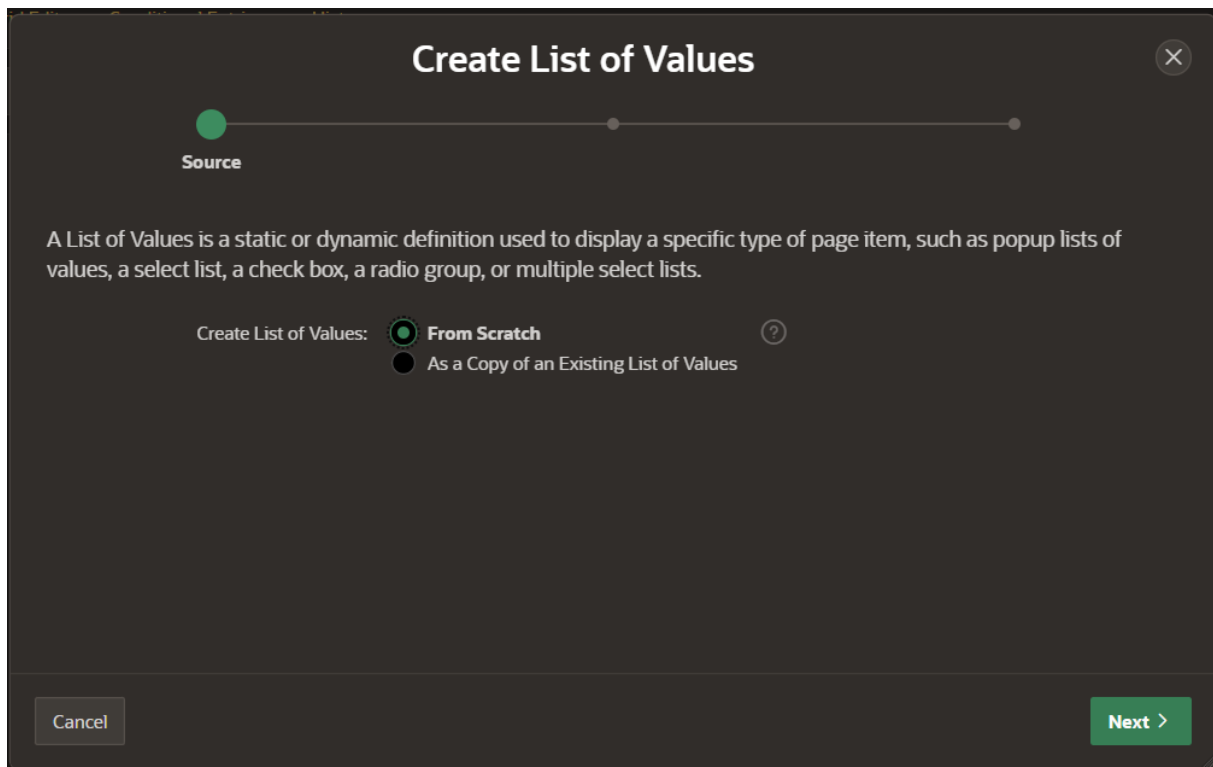
Također su izvršene i neke dodatne izmjene nad pojedinim stupcima tablice. To je učinjeno kako bi se, na primjer, umjesto šifre tipa smještaja prikazivao sam naziv tipa smještaja, s obzirom da šifra tipa smještaja korisniku znači puki podatak, a ne korisnu informaciju. Iz tog su razloga stupci SIFRA_TIPA_SMJESTAJA, SIFRA_DRZAVE te POSTANSKI_BROJ definirani kao *Plain Text (based on a List of Values)* kako bi se umjesto tih šifri prikazivali nazivi tipa smještaja, države i mjesta, respektivno. Stoga je najprije bilo potrebno kreirati tri pripadne liste vrijednosti za ove stupce. Naime, lista vrijednosti uspostavlja odnos između dvaju stupaca jedne tablice. Na taj način, primjerice, možemo uspostaviti odnos između stupca koji sadrži šifre tipova smještaja i stupca koji pohranjuje njihove nazive te reći APEX-u da prikaže naziv koji odgovara određenoj šifri tipa smještaja.

U nastavku je prikazan i objašnjen postupak kreiranja liste vrijednosti na primjeru stupca SIFRA_TIPA_SMJESTAJA. S početnog ekrana alata *App Builder* potrebno je kliknuti na veliki gumb *Shared Components*, a potom pod sekcijom *Other Components* odabrati *List of Values* (slika 32).



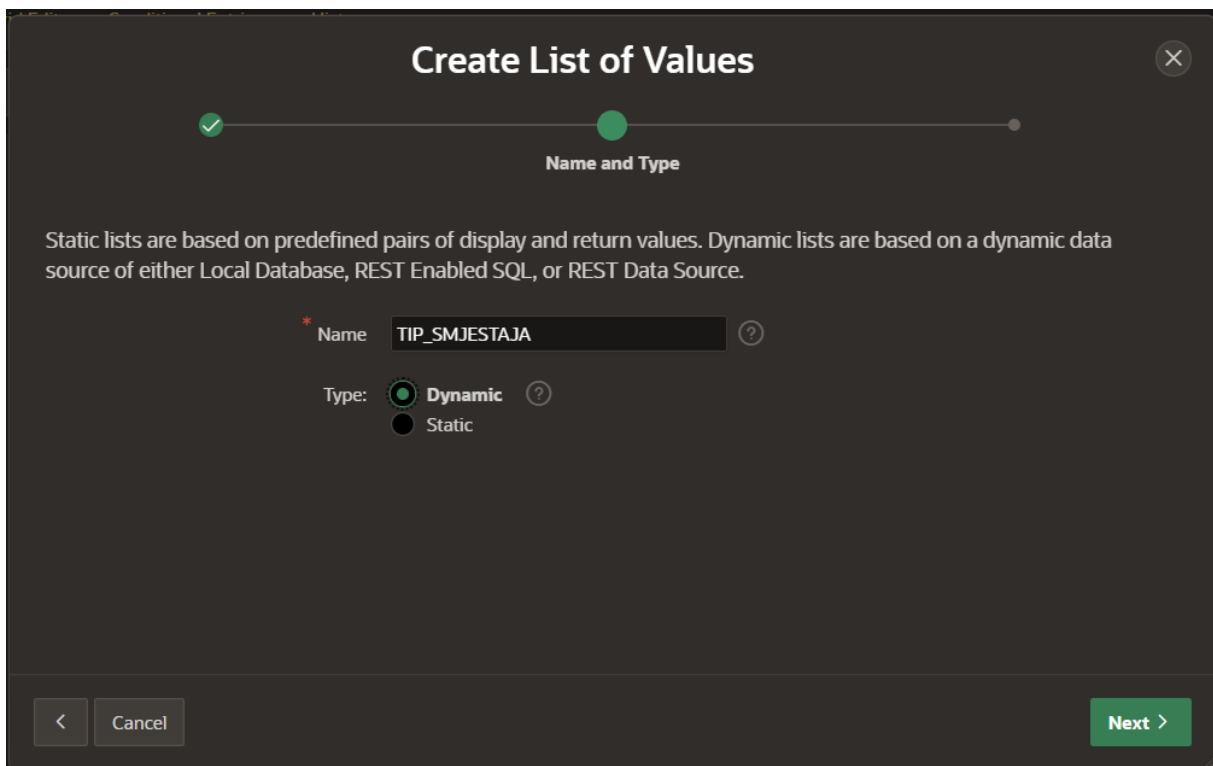
Slika 32. Sučelje alata *Shared Components*

Sada se nalazimo na stranici unutar koje možemo upravljati listama vrijednosti koje postoje za ovu aplikaciju. Nude nam se razne opcije ažuriranja, brisanja ili dodavanja nove liste vrijednosti. Mi trebamo kreirati novu listu, a za to je potrebno kliknuti gumb *Create*, smješten sa lijeve strane zaslona, pri čemu se otvara dijaloški okvir sa slike 33. U prvom koraku čarobnjaka *Create List of Values* odabiremo opciju *From Scratch* (izrađujemo novu listu od nule) te kliknemo na gumb *Next* za prelazak na idući korak.



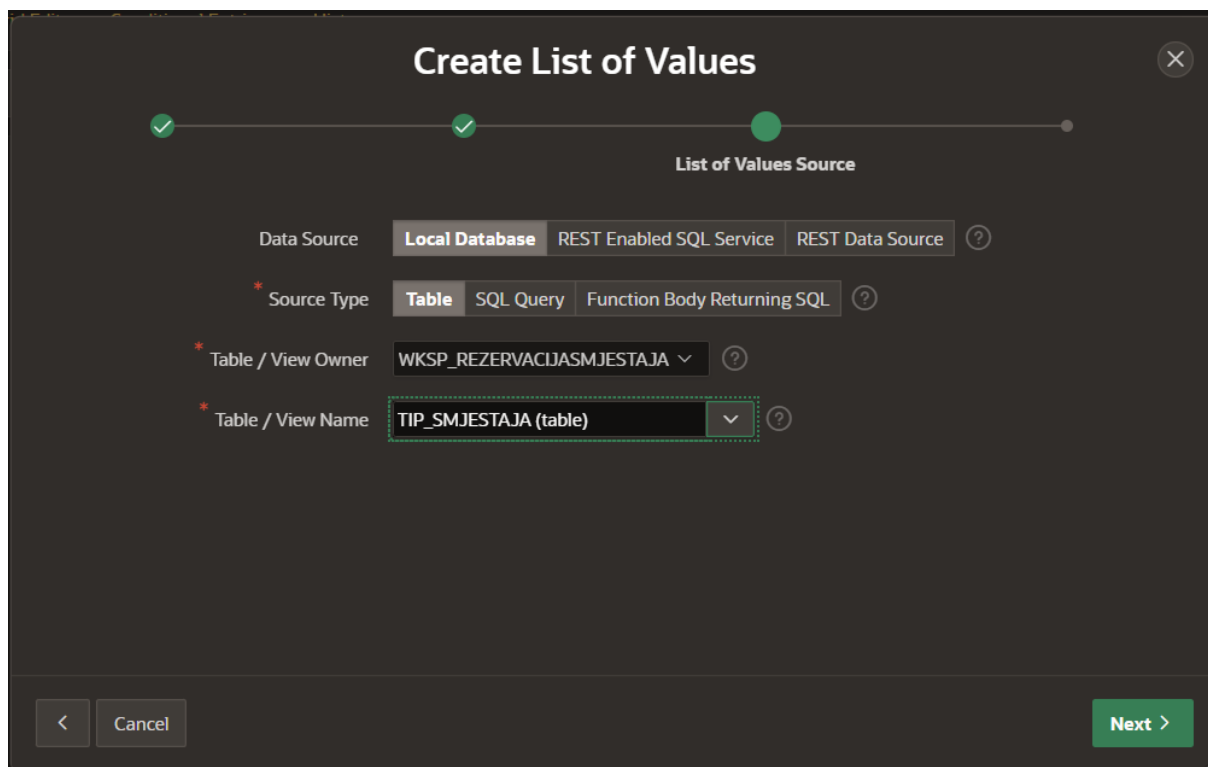
Slika 33. Prvi zaslon čarobnjaka za kreiranje nove liste vrijednosti

U idućem koraku pridružujemo ime ovoj listi – ona je ovdje imenovana kao TIP_SMJESTAJA. Također definiramo tip liste, a u ovom je slučaju riječ o dinamičkoj listi (*Type* → *Dynamic*), s obzirom da želimo podatke dobiti iz baze, a ne ručno upisati vrijednosti u listu. Navedeno je prikazano kroz sliku 34.



Slika 34. Drugi zaslon čarobnjaka za kreiranje nove liste vrijednosti

U idućem je koraku (vidi sliku 35) stoga potrebno definirati izvor iz kojeg dobivamo podatke za listu. Odaberemo da je izvor podataka lokalna baza podataka te odabiremo koja je to tablica iz koje trebamo podatke, u ovom je slučaju to TIP_SMJESTAJA.



Create List of Values

✓ ✓ ●

List of Values Source

Data Source: Local Database REST Enabled SQL Service REST Data Source ?

* Source Type: Table SQL Query Function Body Returning SQL ?

* Table / View Owner: WKSP_REZERVACIJASMJESTAJA ?

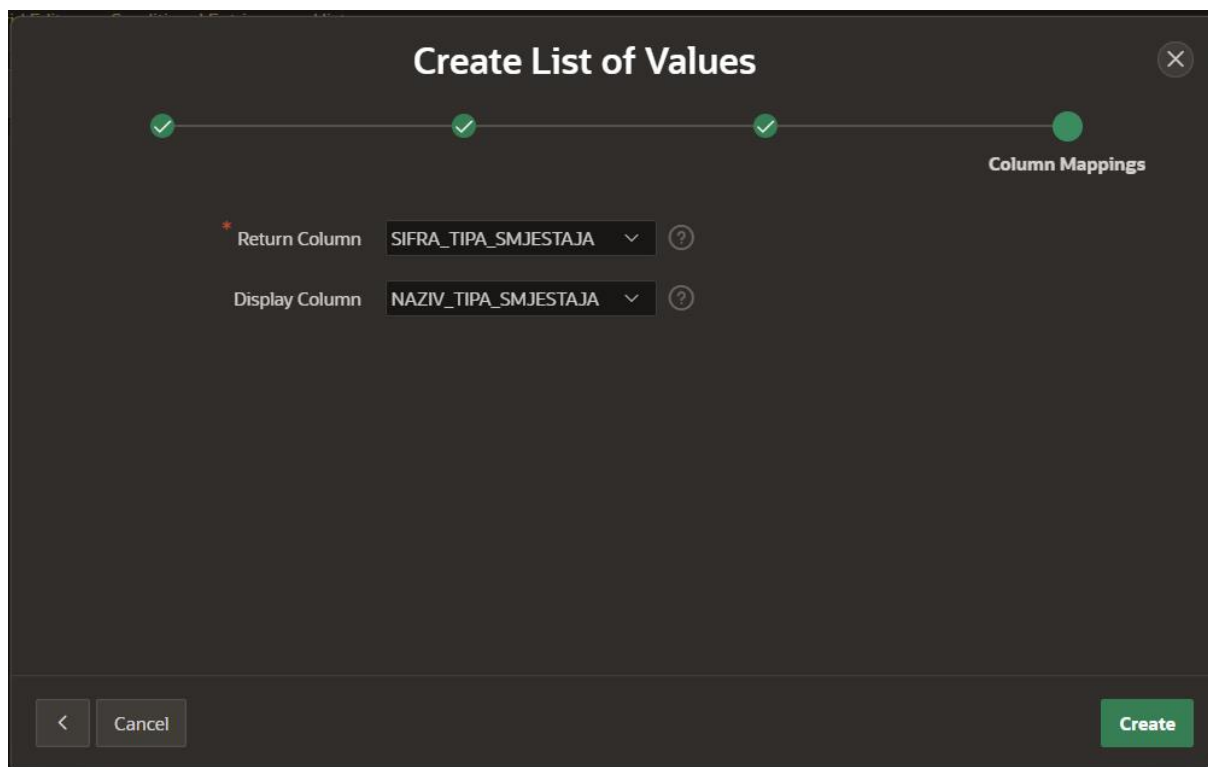
* Table / View Name: TIP_SMJESTAJA (table) ?

< Cancel Next >

Slika 35. Treći zaslon čarobnjaka za kreiranje nove liste vrijednosti

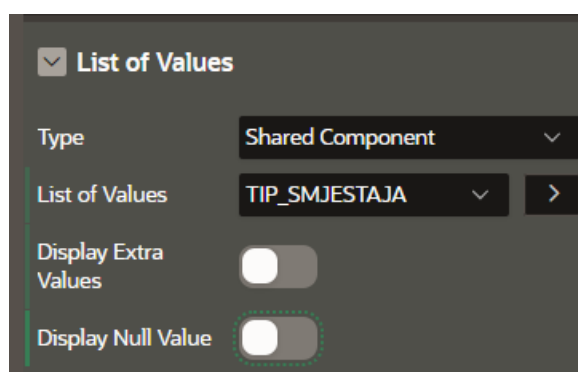
U posljednjem koraku definiramo pridruživanje vrijednosti među stupcima, pri čemu pod polje *Return Column* definiramo koji je to stupac čije vrijednosti želimo vratiti, a pod poljem *Display Column* definiramo stupac čije vrijednosti zapravo želimo ispisati na ekranu (vidi sliku 36).

Ponovimo ekvivalentan postupak za stupce SIFRA_DRZAVE te POSTANSKI_BROJ iz tablica DRZAVA i MJESTO. Prilikom definiranja liste vrijednosti za stupac SIFRA_DRZAVE kao *Display Column* stupac odabiremo NAZIV_DRZAVE, a za stupac POSTANSKI_BROJ pod poljem *Display Column* definiramo stupac NAZIV_MJESTA.



Slika 36. Četvrti zaslon čarobnjaka za kreiranje nove liste vrijednosti

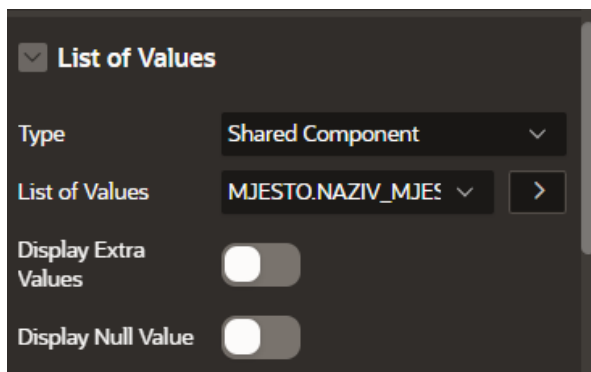
Sada je unutar kreiranog izvješća potrebno stupcu SIFRA_TIPA_SMJESTAJA pridružiti pripadnu listu TIP_SMJESTAJA. Najprije je potrebno odabrati stupac SIFRA_TIPA_SMJESTAJA nakon čega u panelu s desne strane ekrana pod sekcijom *List of Values* jednostavno odaberemo pripadajuću listu vrijednosti nazvanu TIP_SMJESTAJA. Pod *Type* odabiremo *Shared Components*, s obzirom da listu vrijednosti dohvaćamo upravo od tamo (a tamo smo je i maloprije kreirali). Slika 37 prikazuje postavke za sekciju *List of Values* te način na koji je potrebno definirati polja unutar nje. Na isti način pridružujemo liste vrijednosti stupcima SIFRA_DRZAVE ze POSTANSKI_BROJ.



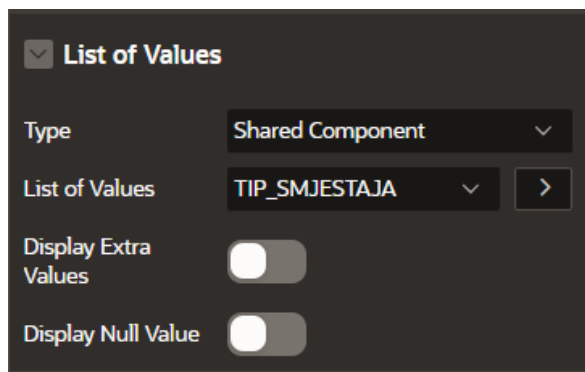
Slika 37. Pridruživanje liste vrijednosti TIP_SMJESTAJA stupcu SIFRA_SMJESTAJA

Kako bi se omogućilo filtriranje smještaja po mjestu i tipu, unutar korisničkog sučelja dodana su dva padajuća izbornika (*Select List*), od koji je prvi nazvan P1_MJESTO_FILTER, a drugi P1_TIP_SMJESTAJA_FILTER. Ove su padajuće liste dodane putem *drag and drop* opcije, na način da je iz panela s dna odabiremo *Items* → *Select List* te istu povlačimo na ekran unutar regije *Filtriranje smještaja*.

Prethodno dodanim padajućim listama potrebno je pridružiti liste vrijednosti koje im pripadaju, a koje se u ovom slučaju odnose na liste putem kojih dohvaćamo naziv mjesta i naziv tipa smještaja, umjesto njihovih šifri. Ovo definiramo unutar desnog panela i sekcije *List of Values*, kako je prikazano sljedećim dvjema zaslona.

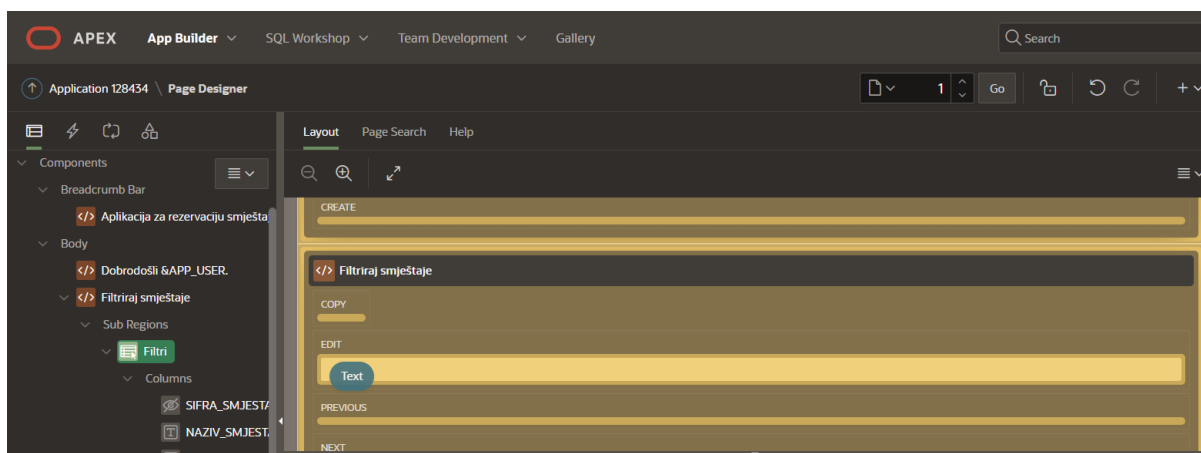


Slika 38. Pridruživanje liste vrijednosti NAZIV_MJEST padajućem izborniku P1_MJESTO_FILTER



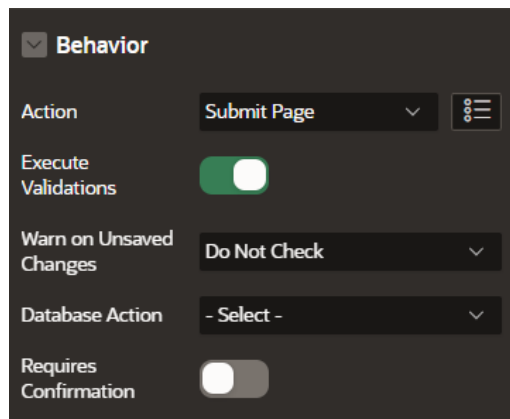
Slika 39. Pridruživanje liste vrijednosti TIP_SMJESTAJA padajućem izborniku P1_TIP_SMJESTAJA_FILTER

Kako bi korisnik uistinu mogao izvršiti filtriranje smještaja, dodan je još i gumb *Filtriraj*, a koji se nalazi na poziciji *Edit* regije *Filtriranje smještaja*. Gumb je dodan na način da smo iz panela smještenog na dnu ekrana odabrali kraticu *Buttons* te zatim na zaslon (unutar *Layout* panela) povukli gumb pod nazivom *Text [Hot]*. Navedeno prikazuje slika 40.



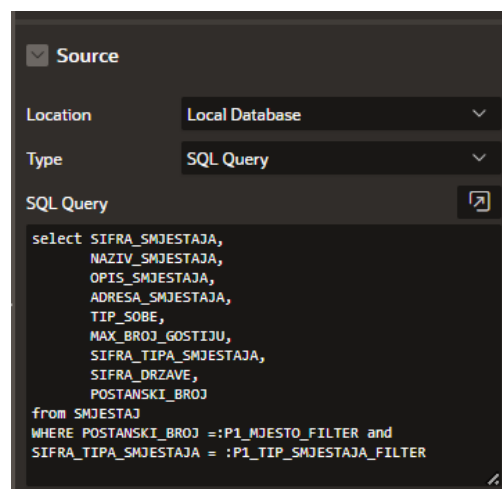
Slika 40. Dodavanje novog gumba pomoću drag and drop opcije

Akcija koju ovaj gumb izvršava jest *Submit Page* pa se tako putem njega izvršavaju postavljeni filteri te korisniku prikazuju samo oni smještaji koji odgovaraju njegovim preferencijama. Postavke gumba definiramo na kartici *Behaviour*, na način da pod poljem *Action* odabiremo iz padajuće liste opciju *Submit Page* (vidi sliku 41).



Slika 41. Definiranje ponašanja i postavki gumba

Posljednji korak kojeg je potrebno izvršiti kako bi filtriranje uistinu davalo željene rezultate odnosi se na promjenu izvora podataka za izvješće *Filtri*. Dosada je kao izvor ovog izvješća služila tablica SMJESTAJ, no sada to postaje SQL upit prikazan slikom 42. Ovaj upit ograničava ispis smještajnih jedinica s obzirom na vrijednosti zapisane unutar polja P1_MJESTO_FILTER te P1_TIP_SMJESTAJA_FILTER.



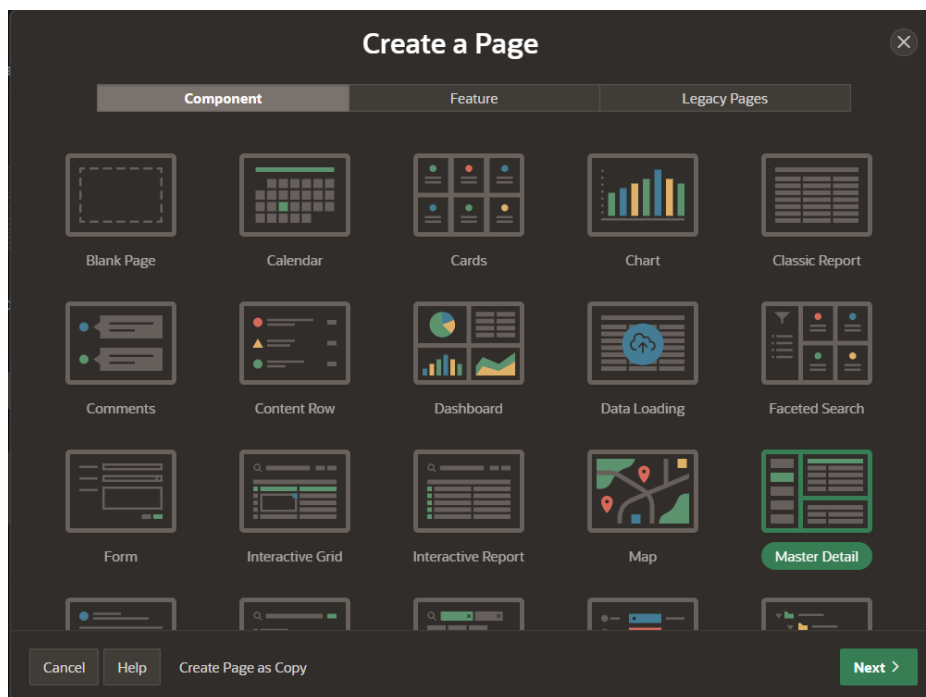
Slika 42. Definiranje izvora podataka izvješća kao rezultata SQL upita

5.5. Master Detail izvješće

Unutar aplikacije kreirana su i Master Detail izvješća, koja se najčešće koriste prilikom prikaza povezanih skupova podataka, na način da je jedan zapis u master skupu povezan s niti jedim ili više zapisa u detail skupu. Stoga je upotreba ovakvih izvješća upravo najprikladniji način prikazivanja agregacija te slabih tipova entiteta. U ovom slučaju je ovakvo izvješće definirano nad tablicama SMJESTAJ i CIJENA, pri čemu se za svaki smještaj prikazuju sve cijene koje su s njim povezane.

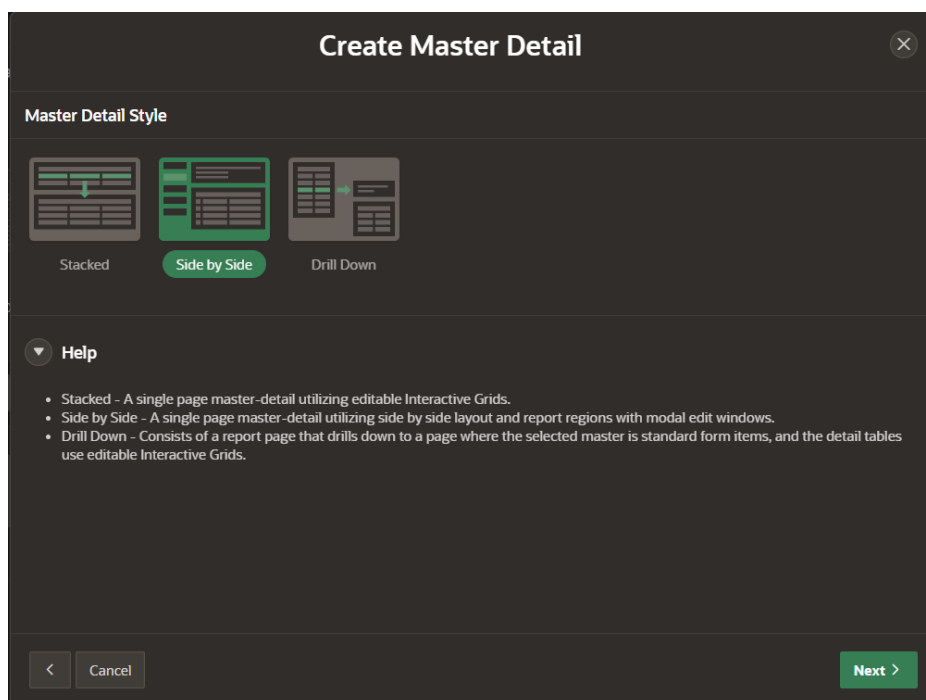
Kako bismo unutar aplikacije dodali novu stranicu, dovoljno je kliknuti gumb *Create Page* smješten na početnom zaslonu same aplikacije. Prvi zaslon ovog čarobnjaka od nas zahtijeva da se odlučimo za tip stranice koji želimo kreirati, kako je prikazano i kroz sliku 43.

Master Detail izvješće možemo definirati odmah prilikom kreiranja nove stranice. Nakon što klinkemo gumb *Create Page*, iz popisa komponenti koji nam se otvara, odaberemo opciju Master Detail, kako je i prikazano slikom 43. Klikom na gumb *Next* prelazimo na sljedeći korak.



Slika 43. Prvi korak prilikom kreiranja Master Detail stranice

Na idućem se ekranu odlučimo za tip Master Detail izvješća kojeg želimo. U ovom je slučaju odabran *Side by Side* stil, koji će u lijevom dijelu ekrana prikazivati zapise iz master tablice, a klikom na jedan od njih prikazuju se dodatni detalji o odabrenom smještaju kao i sve cijene koje su uz njega vezane i to u desnom dijelu ekrana.



Slika 44. Drugi korak prilikom kreiranja Master Detail stranice

Idući je korak definiranje određenih postavki stranice koju kreiramo. Broj stranice možemo ostaviti onakvim kakav je inicijalno zadan, a u polje *Master Detail Page Name* unosimo ime koje želimo da stranica nosi, primjerice *Smještaji*. Sam naziv je moguće kasnije promijeniti. Na ovom je ekranu također definirano kako stranica neće koristiti Breadcrumb navigaciju, no da ulazi u glavnu navigaciju kao novi element navigacije. Također je i odabrana ikona koja će reprezentirati ovu stranicu unutar navigacije. Način na koji je potrebno ispuniti sva polja ovog zaslona prikazan je slikom 45.

Slika 45. Treći korak prilikom kreiranja Master Detail stranice

Idući se ekran ovog čarobnjaka (slika 46) odnosi na definiranje konkretnih postavki Master dijela izvješća. Odabiremo se za tablicu koja će služiti toj svrsi – u ovom je slučaju riječ o tablici SMJESTAJ, nakon čega biramo dva stupca čije će se vrijednosti prikazivati u lijevom dijelu zaslona te time predstavljati svaki pojedini smještaj. Ovdje su odabrani stupci NAZIV_SMJESTAJA te ADRESA_SMJESTAJA, s obzirom da oni najbolje reprezentiraju pojedini smještaj te zajedno daju najviše informacija o istome. Treba napomenuti da je prilikom kreiranja tablice SMJESTAJ postavljeno ograničenje jedinstvenosti nad poljem NAZIV_SMJESTAJA, kako bi se osiguralo da dva smještaja ne nose isto ime. Ovo je potrebno napraviti s obzirom da će se na više mjesta unutar aplikacije umjesto šifre smještaja koristiti njegov naziv, pa bi dva smještaja s istim nazivom mogla zbuniti korisnika. Pod poljem *Detail Table* izaberemo tablicu CIJENA te klikom na gumb *Create Page* finaliziramo kreiranje ovoga izvješća.

Create Master Detail

* Table / View Owner: WKSP_REZERVACIJASMJESTAJA

* Master Table: SMJESTAJ

* Primary Display Column: NAZIV_SMJESTAJA (Varchar2)

* Secondary Display Column: ADRESA_SMJESTAJA (Varchar2)

* Detail Table: CIJENA

Detail Table 2: - Select Value -

< Cancel Advanced Create Page

Slika 46. Posljednji korak prilikom kreiranja Master Detail stranice

Na ovaj način, korisnik klikom na svaki od smještaja vidi detalje o tom smještaju, kao i sve cijene koje se na njega odnose ili su se odnosile u prethodnim periodima. Kako bi se sam ispis podataka o smještaju i njegovim cijenama približio korisnicima, izvršeno je nekoliko sitnih promjena nad stupcima ovih izvješća. Svakom je stupcu nadjenut naziv koji je pristupačniji korisnicima aplikacije te su stupci SIFRA_TIPA_SMJESTAJA, SIFRA_DRZAVE i SIFRA_MJESTA definirani kao *Plain Text (based on List of Values)* kako bi se umjesto šifri prikazivali konkretni naziv, a čime smo se već prethodno upoznali. Nazivi stupaca preimenovani su i kod tablice CIJENA. Također, izmijenjen je redoslijed stupaca za master izvješće SMJESTAJ kako bi njihov raspored imao logičniji slijed. Nadalje, gumbi *Edit*, *Reset* i *Create* prevedeni su na hrvatski jezik kako bi odgovarali jeziku ostatka aplikacije.

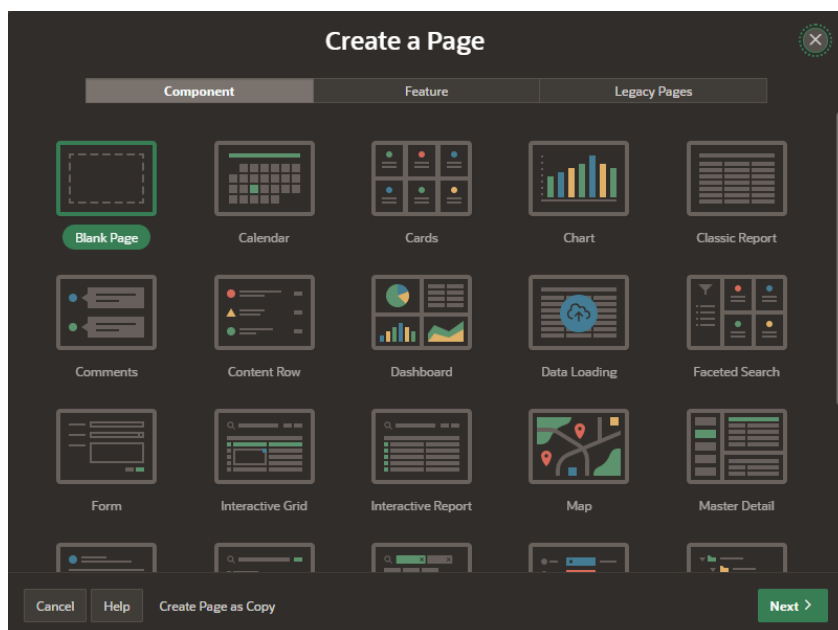
Prilikom kreiranja ovog izvješća, automatski je kreirana i forma SMJESTAJ putem koje je moguće unijeti novi smještaj u bazu podataka. Unutar ove forme su također uvedene izmjene na način da se tip smještaja, država i mjesto biraju iz pripadnih padajućih listi, a ne unose ručno od strane korisnika. Već je prethodno navedeno kako ovo definirati pa ovdje neće biti ponovno opisivano.

Uz ovu formu, automatski je nastala i forma nazvana CIJENE, a koja omogućuje unos nove cijene za pojedinu smještajnu jedinicu. Unutar nje, stupac SIFRA_SMJESTAJA definiran je kao Select List kako bi korisnik jednostavno odabrao smještaj za koji želi unijeti cijenu, bez da zna njegovu šifru. Također, preimenovane su oznake stupaca koje se prikazuju korisniku te prevedeni gumbi kojima se izvršavaju sve potrebne akcije unutar ove forme.

5.6. Interaktivne mreže (Interactive Grids)

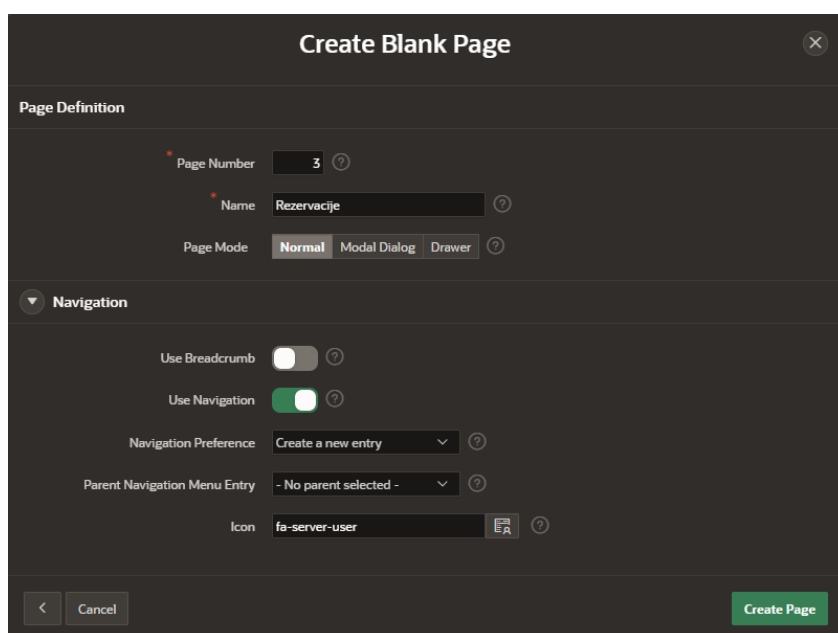
5.6.1. Rezervacije i plaćanja

Najprije ćemo kreirati stranicu koja će prikazivati popis svih rezervacija koje postoje unutar baze podataka. U tu je svrhu najprije potrebno kreirati samu stranicu koja će sadržavati potrebno izvješće. Stranicu dodajemo kao i dosada, klikom na gumb *Create Page*. U ovom slučaju iz popisa komponenti odabiremo *Blank Page* – sama interaktivna mreža (*Interactive Grid*) bit će naknadno dodana na stranicu. Regije interaktivnih mreža omogućuju direktnije, jednostavnije i brže dodavanje, ažuriranje te brisanje podataka iz baze podataka (za razliku od izvješća).



Slika 47. Kreiranje nove prazne stranice

U idućem koraku ovog čarobnjaka unosimo naziv nove stranice – *Rezervacije* te definiramo kako će ova nova stranica činiti dio navigacije web aplikacije (slika 48).



Slika 48. Definiranje postavki stranice

Dodajmo sada na stranicu novu regiju koja će biti tipa *Interactive Grid* i nadjenimo joj ime *Rezervacije*. Cilj ove regije jest prikazati popis svih rezervacija povezanih s prijavljenim korisnikom, tako da svaki korisnik vidi samo svoje rezervacije. Iz tog razloga, kao izvor podataka ove interaktivne mreže koristimo SQL upit prikazan slikom 49. Ovaj upit dohvaća podatke o rezervacijama i korisnicima putem zajedničkog OIB-a te filtrira rezultate tako da prikazuje popis rezervacija samo onog korisnika čije korisničko ime odgovara trenutno prijavljenom korisniku (pritom ignorira mala i velika slova u korisničkom imenu).

The screenshot shows the 'Define Region' dialog in Oracle APEX. The 'Identification' section has 'Title' set to 'Rezervacije' and 'Type' set to 'Interactive Grid'. The 'Source' section has 'Location' set to 'Local Database' and 'Type' set to 'SQL Query'. The SQL query is as follows:

```
select REZERVACIJA.SIFRA_SMJESTAJA,
       REZERVACIJA.OIB,
       REZERVACIJA.DATUM_PRIJAVE,
       REZERVACIJA.DATUM_ODJAVE,
       REZERVACIJA.BROJ_MOCENJA,
       REZERVACIJA.UKUPNA_CIJENA,
       KORISNIK.KORISNICKO_IME
from REZERVACIJA
join KORISNIK ON REZERVACIJA.OIB = KORISNIK.OIB
where UPPER(KORISNIK.KORISNICKO_IME) = UPPER(:APP_USER)
```

At the bottom, there are fields for 'Page Items to Submit' and 'Optimizer Hint'.

Slika 49. Definiranje postavki regije *Rezervacije*

Za ovu ćemo interaktivnu mrežu korisniku omogućiti opcije njezinog uređivanja, čime se korisniku nude opcije ažuriranja i brisanja podataka. Kako bismo ovo učinili, odaberemo regiju interaktivne mreže *Rezervacije*, zatim u panelu s desne strane odabiremo karticu *Attributes* te pod sekcijom *Edit* omogućimo opcije ažuriranja i brisanja redaka ove mreže (vidi sliku 50). Kasnije će biti provedeni autorizacijski postupci koji će osigurati da ove operacije budu dozvoljene jedino administratorima aplikacije.

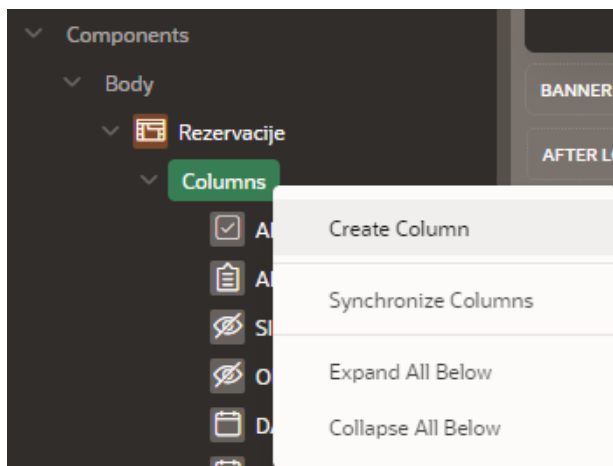
The screenshot shows the 'Attributes' tab in the Oracle APEX 'Edit Region' dialog. The 'Edit' section is expanded, showing the following settings:

- Enabled:** A toggle switch is turned on.
- Allowed Operations:** A list of checkboxes where 'Add Row' is unchecked, and 'Update Row' and 'Delete Row' are checked.
- Allowed Row Operations Column:** A dropdown menu set to '- Select -'.
- Lost Update Type:** A dropdown menu set to 'Row Values'.

Slika 50. Dozvoljavanje operacija ažuriranja i brisanja redaka interaktivne mreže

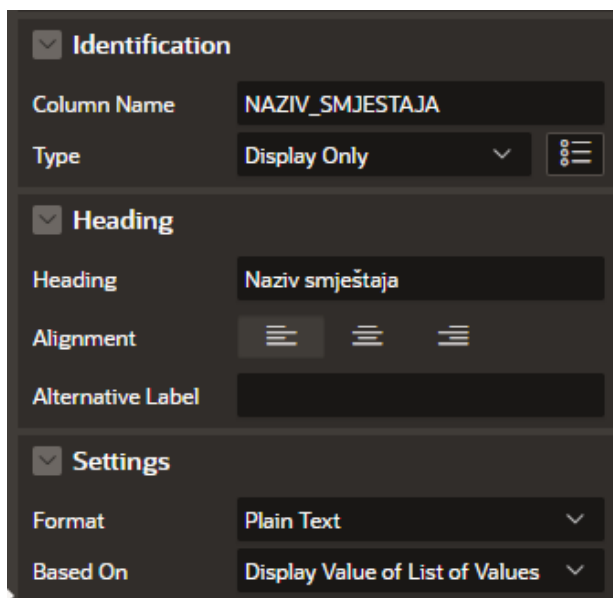
Kako bismo korisniku olakšali pronalazak željene rezervacije, unutar interaktivne mreže *Rezervacije* potrebno je dodati jedan novi stupac koji će prikazivati naziv smještaja na kojeg se dotična rezervacija odnosi.

U nastavku je opisan postupak dodavanja stupca za prikaz naziva rezerviranog smještaja unutar navedene interaktivne mreže. Novi stupac dodajemo putem hijerarhijskog drveća stranice smještenog u panelu s lijeve strane: *Rezervacije* → *Columns* → *Create Column*.



Slika 51. Dodavanje novog stupca

Potom u panelu s desne strane imenujemo stupac te pod poljem *Type* definiramo vrstu stupca kao *Display Only*. Vrlo su bitne postavke pod karticom *Settings*, koje je potrebno postaviti na način prikazan slikom 52.



Slika 52. Prvi dio postavki stupca NAZIV_SMJESTAJA

Pod karticom *List of Values* povezujemo ovaj stupac s pripadajućom listom vrijednosti, koju smo prethodno stvorili putem *Shared Components*. Za izvor podataka ovog stupca pod karticom *Source* odabiremo stupac OIB. Naime, ova interaktivna mreža prikazuje podatke iz tablice REZERVACIJA, što znači da izvor podataka ovog stupca može biti samo jedan od stupaca iz te tablice. Slika 53. prikazuje spomenute postavke.

Slika 53. Drugi dio postavki stupca NAZIV_SMJESTAJA

Kako bismo mogli pregledati sva plaćanja povezana s konkretnom rezervacijom, dodana je nova regija, također tipa *Interactive Grid*, a koja će prikazivati podatke iz tablice PLACANJE. Upravo je ova funkcionalnost razlog korištenja interaktivne mreže za prikaz podataka iz tablice REZERVACIJA (umjesto korištenja interaktivnog izvješća). Naime, interaktivne mreže omogućuju nam definiranje master-detail povezanosti među dvjema mrežama, dok to nije slučaj kod interaktivnih izvješća.

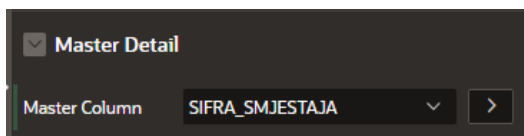
Dodajmo najprije na stranicu još jednu regiju koja je tipa *Interactive Grid*. Kao izvor podataka definiramo tablicu PLACANJE: *Source* → *Type: Table/View* → *Table Name: PLACANJE*. Naziv ove regije definiran je kao *Plaćanja*.

Kako bismo ove dvije mreže povezali putem master-detail odnosa, odaberimo putem panela s lijeve strane regiju *Plaćanje*. Sada u njezinim postavkama (unutar panela s desne strane), pod karticom *Master Detail*, postavljamo regiju *Rezervacije* kao *Master Region* (vidi sliku 54). Ovo implicira da je regija *Plaćanje* stoga *Detail Region* te je time ovisna o regiji *Rezervacije*.

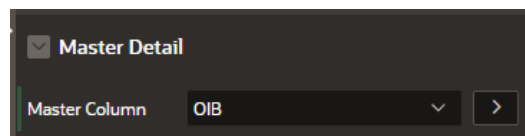
Slika 54. Definiranje master regije za regiju Plaćanje

Zatim stupce SIFRA_SMJESTAJA i OIB iz tablice PLACANJE povezujemo s ekvivalentnim stupcima tablice REZERVACIJA (koji se u ovom slučaju isto nazivaju SIFRA_SMJESTAJA i OIB). Navedeno činimo putem postavke *Master Column*, što možemo vidjeti na slikama 55. i 56. Ovime će se za svaku rezervaciju prikazivati samo ona plaćanja koja su povezana upravo s njom, umjesto da se prikazuju sva plaćanja prisutna u cjelokupnoj tablici. Drugim riječima,

klikom na jedan redak podataka iz interaktivne mreže *Rezervacije*, u mreži *Plaćanja* prikazuju se samo ona plaćanja koja se odnose na tu konkretnu rezervaciju, to jest prikazuju se ona plaćanja čije su vrijednosti stupaca SIFRA_SMJESTAJ i OIB jedanke vrijednostima tih stupaca u tablici REZERVACIJA.



Slika 55. Definiranje Master stupca za stupac SIFRA_SMJESTAJ tablice PLACANJE



Slika 56. Definiranje Master stupca za stupac OIB tablice PLACANJE

S obzirom na to da je korisnicima potrebno dozvoliti unos novog plaćanja za pojedinu rezervaciju, nužno je osigurati logičku ispravnost unesenih podataka. Konkretno, potrebno je osigurati da iznos prvog unesenog plaćanja bude manji ili jednak ukupnom iznosu rezervacije. Nadalje, iznos drugog unesenog plaćanja (ukoliko ono postoji) mora biti jednak razlici između ukupne cijene rezervacije i iznosa prvog plaćanja. Kako bismo implementirali navedeno, potrebno je dodati novi okidač (eng. *trigger*) u bazu podataka, a što možemo učiniti putem alata *SQL Commands*. Upit kojeg je pritom potrebno izvesti prikazan je slikom 57.

```

1 CREATE OR REPLACE TRIGGER provjera_placanja
2 BEFORE INSERT ON PLACANJE
3 FOR EACH ROW
4 DECLARE
5     ukupna_cijena NUMBER;
6     ukupni_iznos NUMBER;
7 BEGIN
8     -- Računanje ukupne cijene rezervacije
9     SELECT Ukupna_cijena
10    INTO ukupna_cijena
11   FROM REZERVACIJA
12  WHERE SIFRA_SMJESTAJ = :NEW.SIFRA_SMJESTAJ
13        AND OIB = :NEW.OIB;
14
15     -- Provjera uvjeta
16     IF ukupna_cijena IS NULL THEN
17         RAISE_APPLICATION_ERROR(-20001, 'Rezervacija ne postoji.');

```

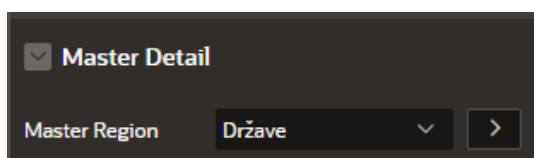
Slika 57. SQL upit za kreiranje okidača "provjera_placanja"

Naposlijetku, kako bi se poboljšalo korisničko sučelje i korisničko iskustvo, odnosno kako bi stranica *Rezervacije* izgledala preglednije, oko dviju interaktivnih mreža dodane su regije statičnog sadržaja kojima su nadjenuti prikladni nazivi – *Rezervacije* i *Plaćanja*. Također, unutar mreže koja prikazuje PLAĆANJA, stupac SIFRA_NACINA_PLACANJA (do sada je to bio skriveni stupac) definiran je kao *Select List* kako se onda putem odgovarajuće liste vrijednosti korisniku prikazivao konkretan naziv načina plaćanja, umjesto njegove šifre.

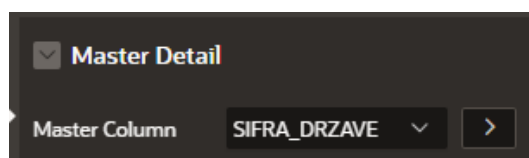
5.6.2. Države i mjesta

Dodana je i stranica za pregled, unos, ažuriranje te brisanje država i mjesta. Ova je stranica definirana na sličan način kao i stranica koja prikazuje rezervacije te plaćanja povezana s njima. Najprije je dodana nova prazna stranica (*Blank Page*), a potom su unutra nje dodane dvije nove regije (*Države* i *Mjesta*) definirane kao *Interactive Grid*.

Ponovno želimo uspostaviti *Master Detail* odnos među ovim tablicama, kako bismo klikom na pojedinu državu dobili popis svih mjesta koja se u njoj nalaze. Stoga je regija *Države* definirana kao *Master Region*: odaberemo regiju *Mjesta* → *Master Detail* → *Master Region* → *Države* (slika 58). Potom za stupac *SIFRA_DRZAVE* definiramo stupac iz tablice *DRZAVA* koji služi kao master stupac putem kojega kreiramo vezu među ovim dvjema tablicama (slika 59).



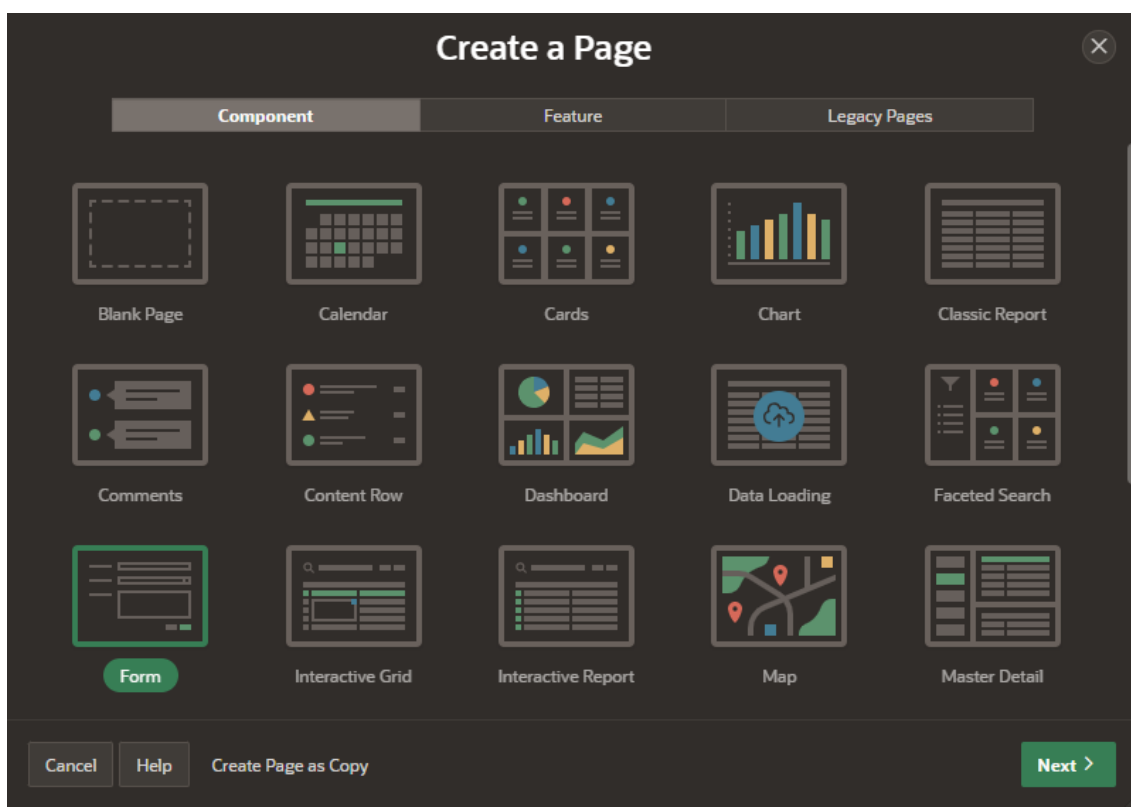
Slika 58. Definiranje Master regije



Slika 59. Definiranje Master stupca

5.7. Forme za unos podataka (Form)

Kako bi se korisniku omogućio unos nove rezervacije za željeni smještaj, kreirana je nova stranica koja će sadržavati formu za unos podataka o novoj rezervaciji. Prilikom kreiranja nove stranice iz galerije komponenti odabiremo *Form* (slika 60).



Slika 60. Kreiranje nove stranice kao forme

U idućem koraku unosimo naziv nove stranice – *Nova rezervacija* (broj stranice možemo ostaviti onakvim kakvim ga APEX automatski definira) te odabiremo tablicu koja će biti izvor podataka → REZERVACIJA. Kod navigacije je ovdje potrebno definirati postavke s kakvim se ranije nismo susreli, s obzirom da ovu stranicu želimo definirati kao „dijete“ stranice REZERVACIJE. Dakle, pod poljem *Parent Navigation Menu Entry* iz padajućeg izbornika odabiremo stranicu *Rezervacije*. Na ovaj će način stranica *Nova rezervacija* biti hijerarhijski niže od njoj nadređene stranice *Rezervacije*. Slika 61. prikazuje na koji je način potrebno ispuniti zaslon čarobnjaka kojim kreiramo formu naziva *Nova rezervacija*.

Create Form

Page Definition

* Page Number: 4

* Name: Nova rezervacija

Page Mode: Normal | Modal Dialog | Drawer

Data Source

Data Source: Local Database | REST Enabled SQL Service | REST Data Source

Source Type: Table | SQL Query

* Table / View Owner: WKSP_REZERVACIJASMJESTAJA

* Table / View Name: REZERVACIJA

Navigation

Use Breadcrumb: ☐

Use Navigation: ☒

Navigation Preference: Create a new entry

Parent Navigation Menu Entry: Rezervacije

Icon: fa-plus

< Cancel Next >

Slika 61. Kreiranje forme *Nova rezervacija*

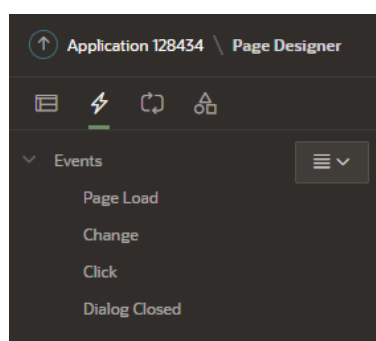
U posljednjem koraku definiramo primarni ključ ove forme, koji nam je već otprije poznat, a čini ga kombinacija stupaca SIFRA_SMJESTAJA i OIB. Unutar oba polja sekcije *Branch Pages* postavljamo vrijednost 3 (odnosi se na broj stranice *Rezervacije*), s obzirom da je to stranica na koju se želimo vratiti ukoliko unesemo novu rezervaciju ili pak odustanemo od unosa nove rezervacije (slika 62). Klikom na gumb *Create Page* završavamo s ovim postupkom.

Slika 62. Posljednji korak prilikom kreiranja nove forme

Kako bismo korisnicima olakšali proces unosa nove rezervacije, implementirat ćemo padajuću listu (*Select List*) za polje SIFRA_SMJESTAJA, koja će sadržavati nazive smještaja umjesto njihovih šifri. Na ovaj način, od korisnika se ne zahtjeva poznavanje šifre smještaja kojeg želi rezervirati, što će učiniti proces rezervacije jednostavnijim. Također, umjesto da od korisnika zahtijevamo unos OIB-a prilikom stvaranja nove rezervacije, koristit ćemo padajuću listu koja sadržava vrijednosti pohranjene u stupcu KORISNICKO_IME. Korisnici će odabrati svoje korisničko ime iz liste, što je bolje rješenje od imena i prezimena jer korisnička imena imaju definirano svojstvo jedinstvenosti u bazi podataka, dok isto ime i prezime mogu pripadati različitim korisnicima.

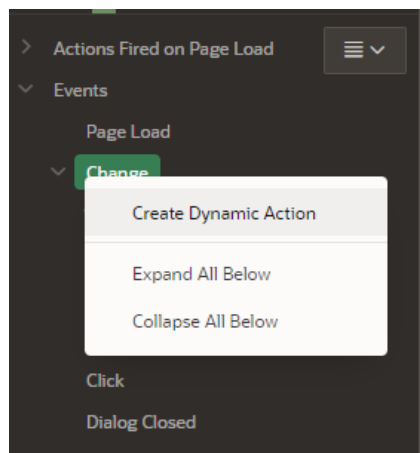
Nadalje, potrebno je izvršiti i određene izmjene nad stupcima BROJ_NOCENJA i UKUPNA_CIJENA. Naime, ne želimo da korisnik sam unosi broj noćenja, već se to polje treba automatski izračunati prilikom unosa datuma prijave i datuma odjave. Također, isto vrijedi i za stupac UKUPNA_CIJENA, gdje želimo da se cijena računa kao umnožak broja noćenja i cijene smještaja kojeg korisnik nastoji rezervirati.

Kako bismo definirali automatsko računanje ovih polja, koristit ćemo se dinamičkim akcijama (*Dynamic Action*), a do kojih dolazimo putem panela s lijeve strane, klikom na ikonu munje u izborniku pri vrhu panela (slika 63).



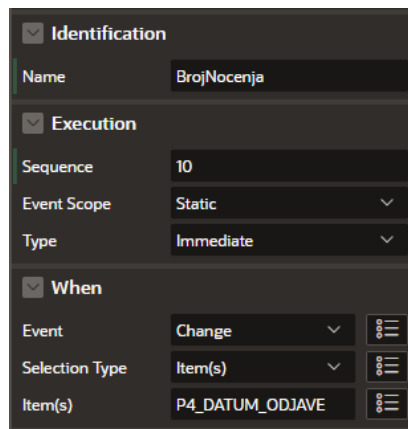
Slika 63. Kartica Dynamic Actions

Novu dinamičku akciju dodajemo desnim klikom miša (pozicioniramo se u ovom slučaju iznad sekcije *Change*) te iz izbornika koji nam se otvara biramo opciju *Create Dynamic Action*, kako je prikazano slikom 64. Time se kreira nova akcija, koja će nam u ovom slučaju javljati greške, s obzirom da je potrebno definirati određene aspekte unutar njezinih postavki.



Slika 64. Kreiranje nove dinamičke akcije

Posvetimo se najprije akciji putem koje će se vršiti automatski izračun broja noćenja nakon unosa datuma prijave i odjave. Unutar postavki za samu dinamičku akciju, najvažnije je definirati postavke pod sekcijom *When*. Ukoliko ovo ne učinimo, dobivat ćemo grešku za ovu dinamičku akciju. U ovom konkretnom slučaju definiramo da će se akcija izvršiti nakon promjene vrijednosti (*Event* → *Change*) varijable P4_DATUM_ODJAVE (slika 65).



Slika 65. Definiranje postavki dinamičke akcije

Potom je potrebno definirati i postavke za *True* granu ove akcije (do nje dolazimo klikom na grešku koja nam se javlja ispod grane *True*, unutar panela s lijeve strane). Ovdje je najvažnije definirati konkretnu akciju koju je potrebno izvršiti, a koja je u ovom slučaju *Set Value* (želimo postaviti vrijednost polju P4_BROJ_NOCENJA). To činimo putem SQL upita pa pod *Set Type* odabiremo *SQL Statement* te u polje za upit naredbe unosimo upit prikazan slikom 66. Pod poljem *Items to Submit* odabiremo P4_DATUM_PRIJAVE, P4_DATUM_ODJAVE te P4_BROJ_NOCENJA. Još je preostalo definirati koji su to elementi na koje ova akcija ima utjecaja, pri čemu se mi tu odabiremo za P4_BROJ_NOCENJA (s obzirom da upravo tom polju želimo postaviti vrijednost putem ove akcije).

Identification

Name:

Action: Set Value

Settings

Set Type: SQL Statement

SQL Statement: `SELECT TO_DATE(:P4_DATUM_ODJAVE) - TO_DATE(:P4_DATUM_PRIJAVE) from dual`

Items to Submit: P4_DATUM_PRIJAVE,P4_D

Escape Special Characters: ☒

Suppress Change Event: ☐

Affected Elements

Selection Type: Item(s)

Item(s): P4_BROJ_NOCENJA

Slika 66. Definiranje postavki za True granu akcije BrojNocenja

Na ovaj je način omogućeno da se korisniku nakon unosa datuma odjave (prije toga i datuma prijave) automatski izračuna broj noćenja, kako ga on ne bi morao samostalno unositi.

Na identičan je način definirano i automatsko izračunavanje vrijednosti polja UKUPNA_CIJENA, koje dobijemo kao umnožak broja noćenja i cijene noćenja. Kako bismo dobili cijenu noćenja, dodan je skriveni stupac unutar kojega smo dohvatili cijenu noćenja s obzirom na šifru smještaja te datuma prijave (koji mora biti između vrijednosti stupaca DATUM_POCEKA i DATUM_ZAVRSETKA) iz tablice CIJENA. Potom je vrijednost tog polja (koje je nazvano kao P4_CIJENA_NOCENJA) pomnoženo s brojem noćenja te se time dobije ukupna cijena rezervacije koju je potrebno platiti (slika 67).

Identification

Name:

Action: Set Value

Settings

Set Type: SQL Statement

SQL Statement: `SELECT :P4_CIJENA_NOCENJA * :P4_BROJ_NOCENJA from dual`

Items to Submit: P4_BROJ_NOCENJA

Escape Special Characters: ☒

Suppress Change Event: ☐

Affected Elements

Selection Type: Item(s)

Item(s): P4_UKUPNA_CIJENA

Slika 67. Definiranje postavki za True granu akcije UkupnaCijena

5.8. Interaktivna izvješća (Interactive Reports)

Sada ćemo definirati jednostavnu stranicu putem koje je moguće vidjeti i uređivati popis svih korisnika. Ove će funkcionalnosti biti omogućene samo administratorima aplikacije, no o tome će biti više riječi kasnije.

Interaktivno izvješće možemo definirati direktno prilikom kreiranja nove stranice: *Create Page* → *Interactive Report*. Kao i do sada, prvi zaslon ovog čarobnjaka od nas zahtijeva unos naziva stranice te definiranje ostalih postavki stranice. Treba napomenuti kako je prilikom kreiranja stranice za interaktivno izvješće automatski kreirana i stranica s formom, potrebna za unos/ažuriranje podataka o korisnicima. Ovo je učinjeno na način da je označena opcija *Include Form Page*, na način prikazan slikom 68. Kao izvor podataka ovog izvješća služit će tablica KORISNIK iz naše baze podataka.

Create Interactive Report

Page Definition

* Page Number: 7 ?

* Name: Korisnici ?

Page Mode: Normal Modal Dialog Drawer ?

Include Form Page: ☒ ?

* Form Page Number: 8 ?

* Form Page Name: Novi korisnik ?

Form Page Mode: Normal Modal Dialog Drawer ?

Data Source

Data Source: Local Database REST Enabled SQL Service REST Data Source ?

Source Type: Table SQL Query ?

* Table / View Owner: WKSP_REZERVACIJASMJESTAJA ?

* Table / View Name: KORISNIK ?

Navigation

Use Breadcrumb: ☐ ?

Use Navigation: ☒ ?

< Cancel Next >

Slika 68. Kreiranje stranice interaktivnog izvješća i njoj pripadne forme za unos podataka

U idućem, a ujedno i posljednjem koraku kreiranja stranice, odabiremo stupac tablice koji služi kao primarni ključ te jedinstveno identificira sve korisnike unutar nje.

Slika 69. Odabir primarnog ključa interaktivnog izvješća

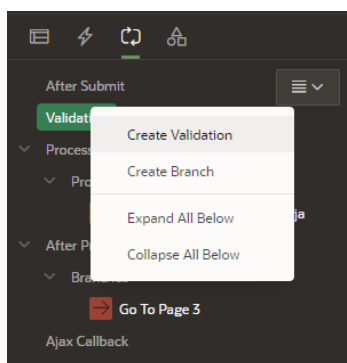
Što se tiče samog interaktivnog izvješća, jedina načinjena promjena odnosi se na preimenovanje njegovih stupaca, kako bi oni bili bliži korisniku. Kod forme za unos novog korisnika načinjeno je nekoliko izmjena. Prilikom unosa podataka o novome korisniku, država i mjesto korisnika definirani su kao padajuće liste (*Select List*), kako se ne bi morala unositi njihova šifra već je dovoljno odabrati nazive države i mjesta koji već postoje u bazi podataka.

5.9. Validacija podataka

Validacija podataka ključan je korak prilikom osiguravanja integracije baze podataka, a najčešće ju provodimo nad ograničenjima, unosima i procesima.

Jedan od primarnih načina osiguravanja točnosti baze podataka jest postavljanje ograničenja nad njezinim tablicama. Ovo je moguće učiniti prilikom kreiranja tablica, ili pak naknadno. Unutar baze podataka nad kojom je kreirana ova aplikacija, već postoje brojna ograničenja, poput ograničenja primarnog i vanjskog ključa, ograničenja jedinstvenosti ili pak ograničenja provjere o tome može li stupac poprimiti vrijednost NULL.

Ukoliko unutar baze podataka pokušamo unijeti neki podatak koji narušava neko od postavljenih ograničenja, dobijemo poruku o pogrešci. Kako bismo mogli postaviti određene provjere podataka i radnji, kreiramo validacije unutar APEX-a. To činimo na način da idemo u *Processing Tab* (treća kartica unutar panela s desne strane), desnim klikom miša odabiremo *Validating* čvor te potom odabiremo *Create Validation*, kako je prikazano kroz sliku 70.



Slika 70. Kreiranje nove validacije

Sada ćemo postaviti provjeru unesenih podataka prilikom unosa nove rezervacije. Naime, ukoliko korisnik pokuša rezervirati smještaj u određenom terminu, a taj je smještaj već prethodno rezerviran od strane nekog drugog korisnika, javlja se poruka o pogrešci koja korisniku nagovještava da taj smještaj nije moguće rezervirati u odabranim datumima.

Najprije dodajemo novu validaciju (na stranici *Nova rezervacija*), kojom ćemo provjeravati uneseni datum prijave, a čije postavke definiramo kako je prikazano slikom 71. Potom dodajemo drugu validaciju, identičnu prvoj, koja se odnosi na provjeru datum odjave (slika 72).

The screenshot shows the 'Validation' configuration panel. The 'Type' is set to 'No Rows returned'. The 'SQL Query' is:


```
SELECT DATUM_PRIJAVE, DATUM_ODJAVE
FROM REZERVACIJA
WHERE :P4_DATUM_PRIJAVE BETWEEN
DATUM_PRIJAVE AND DATUM_ODJAVE
AND SIFRA_SMJESTAJA = :P4_SIFRA_SMJESTAJA;
```

 The 'Always Execute' toggle is turned off. The 'Error Message' is:

Nažalost, nije moguće ostvariti rezervaciju za odabrani termin, jer su datumi već rezervirani. Molimo vas da odaberete druge datume za Vašu rezervaciju.

 The 'Display Location' is set to 'Inline with Field and in N' and the 'Associated Item' is set to '- Select -'.

Slika 71. Validacija za provjeru dostupnosti smještaja s obzirom na datum prijave

The screenshot shows the 'Validation' configuration panel. The 'Type' is set to 'No Rows returned'. The 'SQL Query' is:


```
SELECT DATUM_PRIJAVE, DATUM_ODJAVE
FROM REZERVACIJA
WHERE :P4_DATUM_ODJAVE BETWEEN DATUM_PRIJAVE
AND DATUM_ODJAVE
AND SIFRA_SMJESTAJA = :P4_SIFRA_SMJESTAJA;
```

 The 'Always Execute' toggle is turned off. The 'Error Message' is:

Nažalost, nije moguće ostvariti rezervaciju za odabrani termin, jer su datumi već rezervirani. Molimo vas da odaberete druge datume za Vašu rezervaciju.

 The 'Display Location' is set to 'Inline with Field and in N' and the 'Associated Item' is set to '- Select -'.

Slika 72. Validacija za provjeru dostupnosti smještaj s obzirom na datum odjave

S ciljem osiguravanja veće točnosti same aplikacije, kreirane su i dvije dodatne validacije nad poljima P4_DATUM_PRIJAVE i P4_DATUM_ODJAVE. Ove provjere osiguravaju da su uneseni datumi prijave i odjave ispravni. Konkretno, postavljeni su sljedeći uvjeti: datum prijave ne smije biti stariji ili jednak današnjem datumu, a datum odjave mora vremenski slijediti nakon datuma prijave. Slike 73. i 74. prikazuju postavke ovih validacija, a koje su definirane putem panela s desne strane zaslona.

Validation

Editable Region: - Select -

Type: Expression

Language: SQL

SQL Expression: `:P4_DATUM_PRIJAVE > SYSDATE`

Always Execute: ☐

Error

Error Message: Rezervacije su dostupne samo za buduće datume. Molimo Vas odaberite datum prijave koji pada nakon današnjeg datuma.

Display Location: Inline with Field

Associated Item: P4_DATUM_PRIJAVE

Slika 73. Validacija za provjeru datuma prijave u odnosu na današnji datum

Validation

Editable Region: - Select -

Type: Expression

Language: SQL

SQL Expression: `:P4_DATUM_ODJAVE > :P4_DATUM_PRIJAVE`

Always Execute: ☐

Error

Error Message: Datum odjave mora slijediti nakon datuma prijave. Molimo izaberite ispravne datume za Vašu rezervaciju.

Display Location: Inline with Field

Associated Item: P4_DATUM_ODJAVE

Slika 74. Validacija za provjeru datuma odjave u odnosu na datum prijave

5.10. Autentifikacija i autorizacija korisnika

Sigurnost aplikacije možemo promatrati kroz dva povezana aspekta: autentifikaciju i autorizaciju korisnika. Autentifikacija korisnika identificira „dozvoljene“ korisnike aplikacije, dok autorizacija definira što koji od tih korisnika može vidjeti/učiniti unutar aplikacije.

5.10.1. Autentifikacija

Autentifikacija je dakle mogućnost identifikacije trenutnog korisnika aplikacije. Korisnici APEX-a identificiraju se putem korisničkog imena i lozinke. APEX nudi nekoliko različitih autentifikacijskih shema. Trenutna autorizacijska shema razvijene aplikacije za rezervaciju smještaja jest *Application Express Accounts scheme*, koja omogućuje prijavu samo onim korisnicima koji posjeduju račun za to radno okruženje.

Tri su bitna koraka prilikom provođenja autentifikacije korisnika: pohranjivanje korisničkog imena i lozinke korisnika, upravljanje korisničkim računima te konačna autentifikacija korisnika.

Dakle, prvi je korak pohranjivanje podataka o korisniku, tj. njegovog korisničkog imena i lozinke. U tu je svrhu potrebno kreirati novu tablicu unutar baze podataka – nazovimo ju **USERS** (tablicu korisnik već imamo pa da ne bi došlo do zabune među ovim tablicama). Ova se tablica sastoji od triju stupaca: *UserName*, *Password* te *IsAdmin*. Preporuka je također kreirati i 2 klasična korisnika → jednog s korisničkim imenom *Guest* koji se prijavljuje bez lozinke i koji nije administrator, te jednog administratora koji se prijavljuje korisničkim imenom **ADMIN** i lozinkom **1234**. Kreirajmo spomenutu tablicu i ova dva korisnika putem alata *SQL Commands*, kako je prikazano slikom 75.


```

1 create table USERS (
2     UserName varchar2(12), Password varchar2(1000),
3     IsAdministrator char(1)
4 );
5 insert into USERS (UserName, Password, IsAdministrator)
6     values ('ADMIN', apex_util.get_hash(apex_t_varchar2('ADMIN','1234'), null), 'Y');
7
8 insert into USERS (UserName, Password, IsAdministrator)
9     values ('GUEST', apex_util.get_hash(apex_t_varchar2('GUEST',null), null), 'N');

```

Slika 75. Kreiranje tablice USERS i umetanje podataka u nju

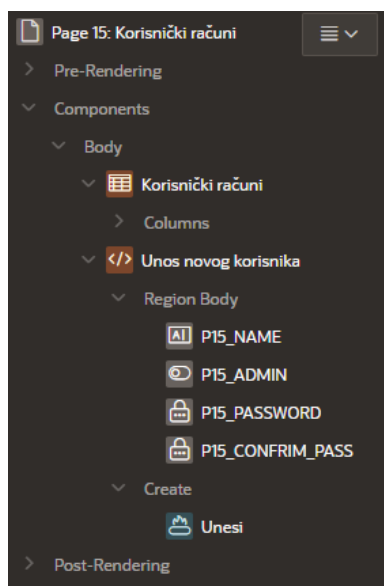
Sigurnosti radi, zaporka su pohranjene u kodiranoj formi (pomoću funkcije *get_hash*), u što se možemo uvjeriti ukoliko ispišemo sve korisnike iz tablice USERS (vidi sliku 76).

USERNAME	PASSWORD	ISADMINISTRATOR
ADMIN	pk3Jic4r3zkJW_6VSSbMq7tznjVvxzj3Ne6Hszc1e1SlgeHr8PLq5zr7qeTcrXSlqv1Zr4Txo9esnhzsUEAA	Y
GUEST	wR8sdanx_8H9-kbbMa6PHEoMWxni5u36PYQorap5pU0zqUywjTlhhgmp7eRlrW8x4iO0IM5FwbbwrO_bNwyQg	N

2 rows returned in 0.02 seconds [Download](#)

Slika 76. Ispis podataka iz tablice USERS

Potom dodajemo novu praznu stanicu kojoj damo naziv *Korisnički računi*. Na toj će se stranici nalaziti klasično izvješće koje će ispisivati podatke iz tablice USERS te regija putem koje će biti moguće unijeti novog korisnika. Nakon umetanja svih potrebnih regija, hijerarhijsko drvo koje prikazuje sve elemente stranice trebalo bi izgledati kako je prikazano slikom 77.

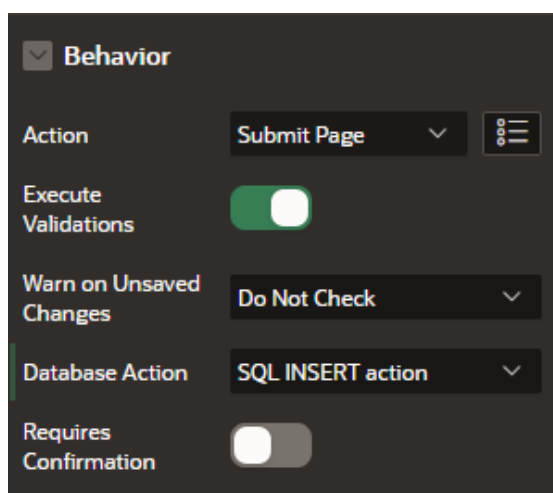


Slika 77. Hijerarhijsko drvo stranice Korisnički računi

Na slici 77. također vidimo da su unutar regije *Unos novog korisnika* dodana 4 elementa (mi ih samostalno dodajemo):

- P15_NAME → koristi se za unos naziva korisničkog računa i tipa je *Text Field*
- P15_ADMIN → koristi se za unos informacije o tome je li korisnik administrator ili ne i tipa je *Radio Group*
- P15_PASSWORD → koristi se za unos lozinke i tipa je *Password*
- P15_CONFIRM_PASS → koristi se za potvrdu unesene lozinke i tipa je *Password*

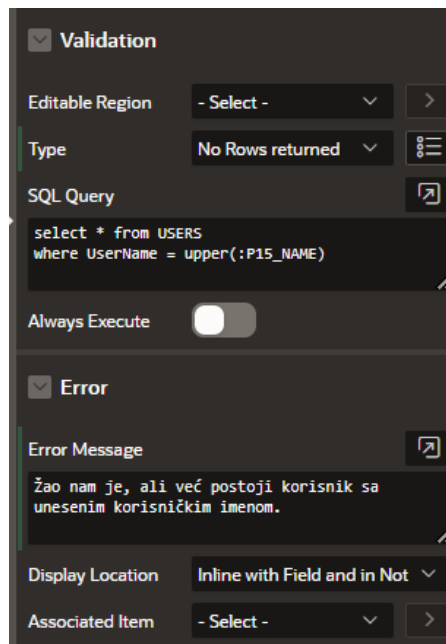
Unutar ove regije i navedenih polja unositi ćemo podatke o korisničkom računu kojeg želimo kreirati. Unutar regije također dodajemo i gumb *Unesi*, koji će služiti za konačan unos novog korisnika u tablicu USERS. Postavke za gumb prikazne su slikom 78.



Slika 78. Postavke za gumb *Unesi* unutar regije *Unos novog korisnika*

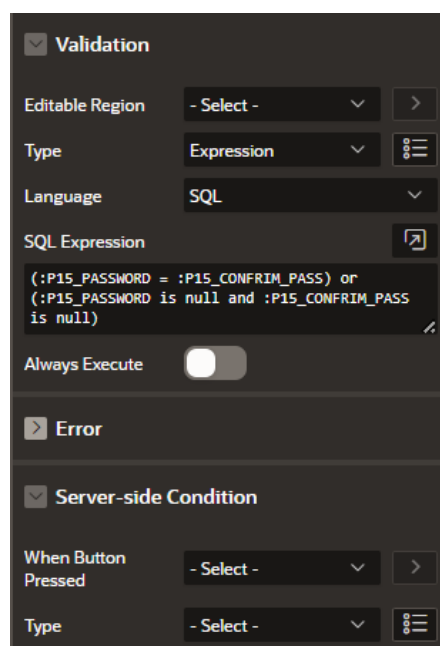
Kako bismo se osigurali da su uneseni podaci ispravni, kreirane su dvije validacije te proces pomoću kojega zapravo i unosimo podatke. Prva validacija provjerava postoji li uneseno korisničko ime u bazi podataka, dok druga provjerava podudaraju li se unesene lozinke. Proces pretvara korisničko ime u sva velika slova, kodira lozinku te umeće zapis u tablicu USERS.

Prva validacija (koja je tipa *No Rows Returned*) definirana je putem SQL upita sa slike 79. Ona provjerava postoji li već unutar tablice USERS korisnički račun s unesenim korisničkim imenom. Ukoliko je to slučaj, onemogućen je unos novog korisnika s tim korisničkim imenom. Te se javlja poruka o pogrešci: *Žao nam je, ali već postoji korisnik s unesenim korisničkim imenom.*



Slika 79. Validacija za provjeru unesenog korisničkog imena

Druga validacija provjerava podudaraju li se unesena lozinka te potvrda lozinke. Ovu je validaciju potrebno definirati na način prikazan slikom 80. Kao i kod prethodne validacije, ukoliko se unesene lozinke ne podudaraju, javlja se poruka o pogrešci i onemogućuje unos novog korisnika.



Slika 80. Validacija za provjeru podudarnosti lozinke

Naposlijetku, kreiran je i proces koji korisničko ime pretvara u velika slova, kriptira lozinku te umeće novog korisnika u tablicu USERS (slika 81). Također su definirane poruke o uspješnosti ili pogrešci izvršavanja procesa. Kako bi gumb bio uistinu funkcionalan, kod procesa moramo definirati postavke pod sekcijom *Server-side Condition* (vidi sliku 82).

Source

Location: Local Database

Language: PL/SQL

PL/SQL Code

```
declare
v_username varchar2(20) := upper(:P15_NAME);
v_valuesToHash apex_t_varchar2 :=
apex_t_varchar2(v_username, :P15_PASSWORD);
begin
insert into USERS (UserName, Password,
IsAdministrator)
values (v_username,
apex_util.get_hash(v_valuesToHash, null),
:P15_ADMIN);
end;
```

Execution

Sequence: 10

Point: Processing

Run Process: Once Per Page Visit (default)

Slika 81. Prvi dio postavki procesa za unos novog korisnika

Success Message

Success Message: Uspješno ste unijeli novi korisnički račun.

Error

Error Message: Došlo je do greške prilikom unosa novog korisničkog računa.

Display Location: Inline in Notification

Server-side Condition

When Button Pressed: Unesi

Type: - Select -

Slika 82. Drugi dio postavki procesa za unos korisnika

Sada kada znamo kako možemo kreirati korisnika, moramo dodati i način na koji ih možemo i autentificirati. U tu je svrhu potrebno kreirati novu autentifikacijsku shemu putem koje će to biti učinjeno. Kako bismo kreirali novu autentifikacijsku shemu, idemo na početnu stranicu uređivača aplikacije, kliknemo *Shared Components* gumb te pod sekcijom *Security* odabiremo *Authentication Schemes*. Kako bismo stvorili novu shemu, kliknemo na gumb *Create*. Na prvom zaslonu čarobnjaka možemo birati želimo li kreirati neku pre-konfiguriranu shemu iz galerije ili pak nastojimo kreirati shemu koja je kopija neke već postojeće autorizacijske sheme. Odaberemo prvu opciju i kliknemo na *Next*.

Create Authentication Scheme

Method

When you create a new authentication scheme, you have several options. Most let you reuse implementations that already exist in your application or in other applications within your workspace. There are even some pretested schemes you can copy to get you up and running immediately.

Create Scheme: ☒ Based on a pre-configured scheme from the gallery ☐ As a copy of an existing authentication scheme

> Information

Cancel Next >

Slika 83. Prvi zaslon čarobnjaka za stvaranje nove autentifikacijske sheme

Na drugom zaslonu unosimo ime ove sheme te biramo njezin tip iz padajućeg popisa dostupnih shema. S obzirom na to za koju se shemu odlučimo, možda će biti potrebno ispuniti i neke dodatne podatke o istoj. Mi odabiremo *Custom* autentifikacijsku shemu, unosimo naziv autorizacijske funkcije u polje *Authentication Function Name* te njezin kod unutar okvira za unos PL/SQL koda, kako je prikazano slikom 84.

Authentication Scheme

Name

* Name: AutentikacijaKorisnika ?

* Scheme Type: Custom ?

Settings

Sentry Function Name: ?

Invalid Session Procedure Name: ?

Authentication Function Name: custom_authentication ?

Post Logout Procedure Name: ?

Enable Legacy Authentication Attributes: No ?

Source

PL/SQL Code ?

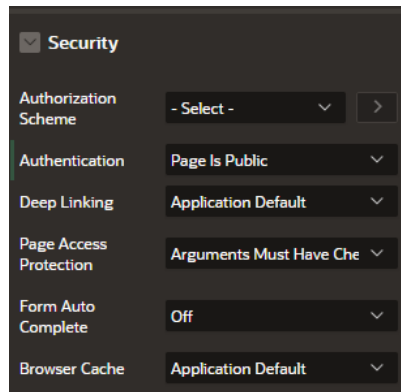
```

1 function custom_authentication (p_username in varchar2, p_password in varchar2) return boolean
2 is
3     v_username varchar2(20) := upper(p_username);
4     v_valuesToHash apex_t_varchar2 := apex_t_varchar2(v_username, p_password);
5     v_userCount int;
6 begin
7     select count(*) into v_userCount
8     from USERS
9     where UserName = v_username
10    and Password = apex_util.get_hash(v_valuesToHash, null);
11
12     return v_userCount = 1;
13 end;
```

Slika 84. Drugi zaslon čarobnjaka za stvaranje nove autentifikacijske sheme

Autentifikacijska funkcija sa slike iznad pretvara sva slova korisničkog imena u velika, provjerava postoji li u tablici *USERS* korisnik s unesenim korisničkim imenom te kodira lozinku. Nakon kreiranja nove sheme, ona automatski postaje aktivnom, a u njezinu ispravnost možemo se uvjeriti ukoliko se pokušamo ulogirati nekim od korisničkih računa koji postoje u tablici *USERS*.

Kako bismo korisnicima koji nisu prijavljeni omogućili da ipak primjerice mogu pregledavati smještaje (no bez mogućnosti rezervacije istih), stranice *Početna* i *Smještaji* bit će definirane kao javno dostupne stranice, putem opcije *Page is Public*. Kako bismo ovo definirali, odemo na uređivače tih stranica, pronađemo u panelu s desne strane sekciju *Security* te pod poljem *Authentication* odabiremo *Page is Public* (vidi sliku 85).



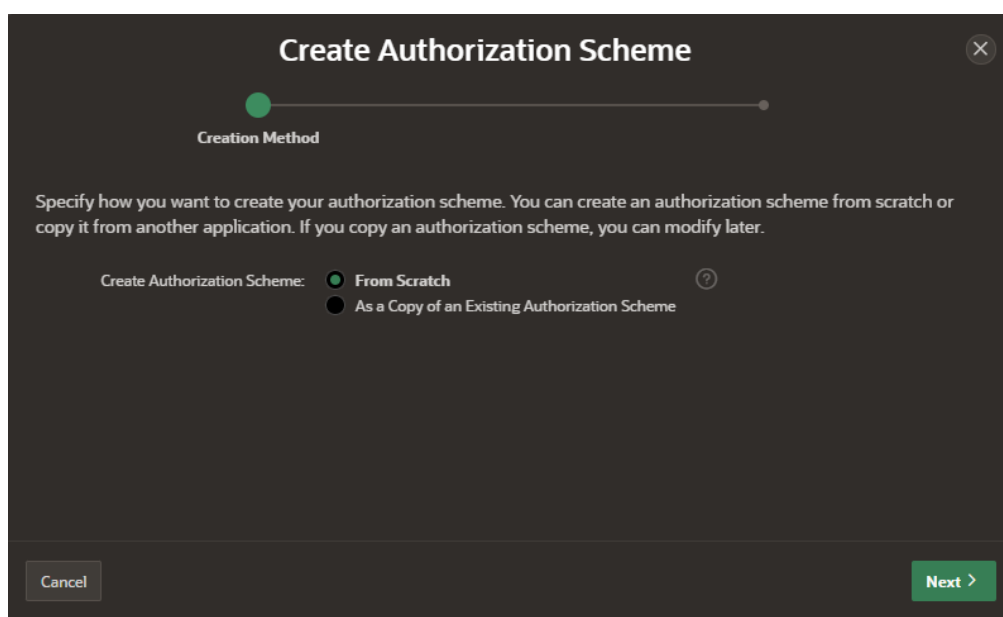
Slika 85. Definiranje stranice kao javno dostupne

5.10.2. Autorizacija

Autorizacija definira kojim korisnicima je dozvoljen pristup stranicama aplikacije koje nisu definirane kao javno dostupne stranice. Također, s obzirom da postoje različiti korisnici aplikacije, pojedinicima od njih je dozvoljen pristup samo nekim dijelovima web aplikacije ili primjerice tek određenim dijelovima pojedinih stranica. APEX implementira autorizaciju na način da svaka komponenta ili stranica ima sa sobom povezanu autorizacijsku shemu koja specificira njoj dozvoljene korisnike.

Ove sheme često uključuju korisničko ime korisnika prijavljenog u aplikaciju, putem već spomenute varijable `APP_USER`. Kreirajmo sada autorizacijsku shemu Administratori, koja nam kazuje o tome ima li korisnik administrativne ovlasti ili ne (je li prijavljen kao ADMIN), shemu GOSTI koja kazuje je li korisnik prijavljen kao GUEST te shemu USERS koja vraća istinu ukoliko je korisnik prijavljen bilo kao GUEST ili kao ADMIN.

Autorizacijska shema kreira se na identičan način kao i autentifikacijska shema: *Shared Components* → *Security* → *Authorization Schemes*. Klikom na gumb *Create* započinjemo s procesom kreiranja nove sheme. Na prvom ekranu ovog čarobnjaka specificiramo da želimo kreirati novu shemu počevši od nule (*From Scratch*), a ne kao kopiju neke već postojeće sheme.



Slika 86. Prvi zaslon čarobnjaka za kreiranje nove autorizacijske sheme

Na idućem zaslonu unosimo ime i tip sheme te unosimo specifikacije za istu. Na slici 87. prikazan je način na koji je potrebno ispuniti ovaj zaslon za autorizacijsku shemu *Administratori*.

Create Authorization Scheme

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: **128434 Aplikacija za rezervaciju smještaja**

Name: **Administratori**

Scheme Type: **Exists SQL Query**

SQL Query: `select *
from USERS
where UserName = :APP_USER and IsAdministrator = 'Y'`

Identify error message displayed when scheme violated: **Pristup je dozvoljen isključivo administratorima aplikacije!**

Validate authorization scheme: ☒ **Once per session**
☐ Once per page view
☐ Once per component
☐ Always (No Caching)

< Cancel **Create Authorization Scheme**

Slika 87. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme *Administratori*

Kreirajmo sada i preostale dvije autorizacijske sheme (*Gosti* i *Built-in korisnici*) na način prikazan slikama 88. i 89.

Create Authorization Scheme

✕

✓
Details

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: **128434 Aplikacija za rezervaciju smještaja** ?

* Name: ?

* Scheme Type: PL/SQL Function Returning Boolean ?

* PL/SQL Function Body

```
return :APP_USER = 'GUEST';
```

?

Identify error message displayed when scheme violated

Pristup dozvoljen gostima aplikacije.

?

Slika 88. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme Gosti

Create Authorization Scheme

✕

✓
Details

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: **128434 Aplikacija za rezervaciju smještaja** ?

* Name: ?

* Scheme Type: PL/SQL Function Returning Boolean ?

* PL/SQL Function Body

```
return :APP_USER = 'GUEST' or :APP_USER = 'ADMIN';
```

?

Identify error message displayed when scheme violated

Pristup dozvoljen grupama korisnika Gosti i Administratori

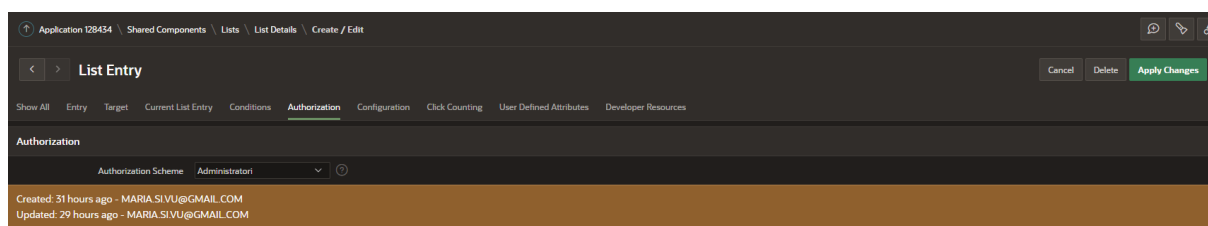
?

Slika 89. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme Built-in korisnici

Sada je potrebno kreirane autorizacijske sheme pridružiti konkretnim komponentama aplikacije. Svaka komponenta pod sekcijom *Security* sadrži postavku za polje *Autorization Sheme*, putem koje možemo iz padajuće liste odabrati kojim je to korisnicima dozvoljen pristup

toj komponenti. Autorizacijsku shemu možemo dodijeliti i cijeloj stranici pa na taj način ona uopće neće biti vidljiva korisnicima koji ne pripadaju toj shemi. Ukoliko korisnik koji nije jedan od onih kojima je pristup dozvoljen, korisniku se prikazuje poruka o pogrešci i onemogućuje mu se ulazak na tu stranicu. Kako bismo poboljšali iskustvo korištenja aplikacije, najbolje je takve stranice u potpunosti sakriti od onih korisnika kojima nije ovlašten pristup istima (primjerice, takvim korisnicima se ta stranica neće prikazivati niti unutar navigacije).

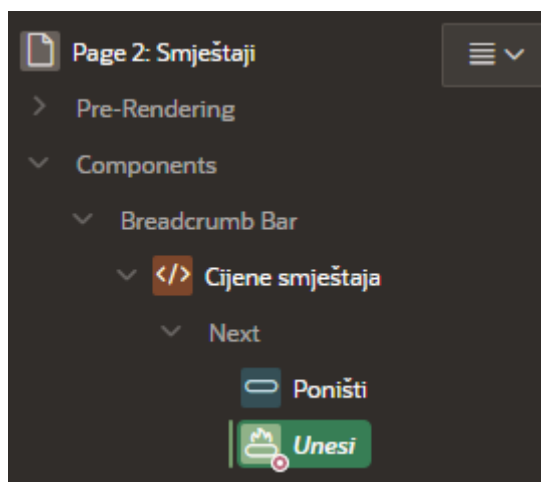
Kako bismo određenu stranicu sakrili iz navigacije od određenih korisnika, pratimo sljedeće korake. Idemo na *Shared Components* → *Navigation and Search* → *Navigation Menu* → tražimo stranicu nad kojom želimo uspostaviti autorizaciju. Potom odlazimo na karticu *Authorization* te biramo onu shemu koja je potrebna. Primjer je dan kroz sliku 90.



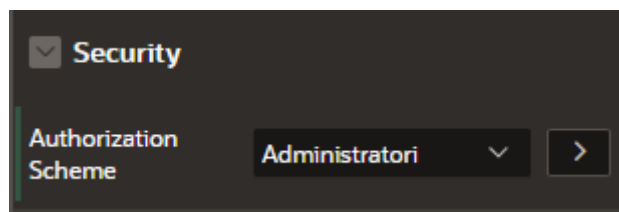
Slika 90. Postavljanje autorizacije nad stranicom navigacijskog izbornika

U nastavku slijedi opis postavljanja autorizacijskih shema za konkretne stranice i komponente aplikacije. Kao što je već rečeno, stranice *Početna* i *Smještaji* definirane su kao *Page is Public* stranice kojima mogu pristupiti i neprijavljeni korisnici. Nad stranicom *Početna* ne unosimo nikakve dodatne promjene. Nad stranicom *Smještaji* dodatno ćemo definirati da je unos novog smještaja i novih cijena omogućen samo administratorima aplikacije, dok svi drugi korisnici imaju mogućnost pregledavanja ovih podataka. Isto činimo i za gumb *Uredi* kojim je moguće uređivanje podataka o smještaju te za gumb putem kojeg se dodaju nove cijene smještaja.

Evo primjera kako dodati autorizacijsku shemu nad gumbom *Unesi* (ekvivalentan postupak provodi se i prilikom dodavanja autorizacijske sheme za bilo koju komponentu web aplikacije). Najprije odabiremo željeni element iz hijerarhijskog drva putem panela s lijeve strane zaslona (slika 91). Potom prelazimo u desni panel unutar kojeg pronalazimo sekciju *Security* te definiramo željenu autorizacijsku shemu pod poljem *Authorization Scheme* (vidi sliku 92). U ovom slučaju riječ je o shemi *Administratori* koju smo maloprije samostalno kreirali.

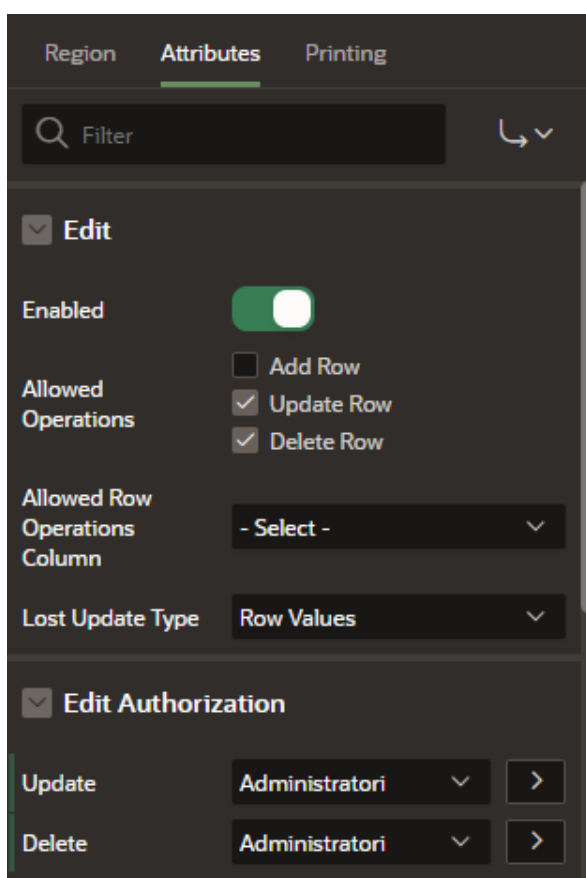


Slika 91. Odabir komponente nad kojom želimo postaviti autorizacijsku shemu

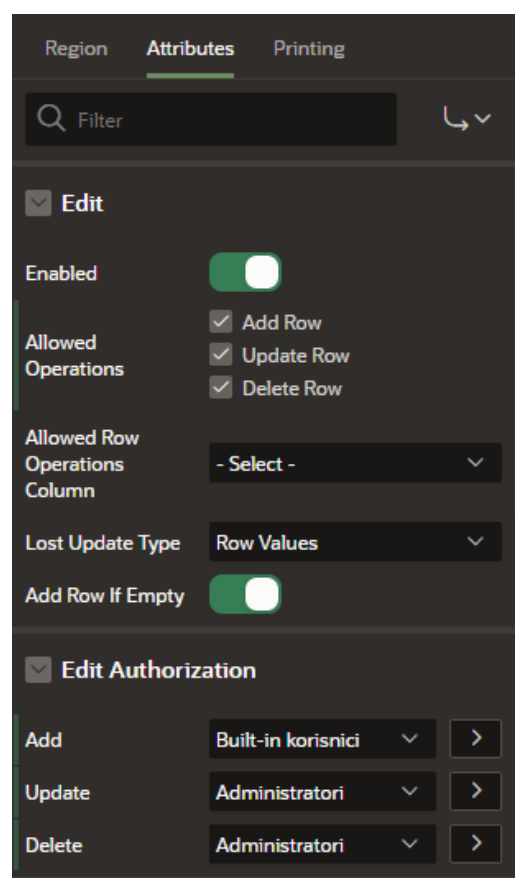


Slika 92. Definiranje autorizacijske sheme za odabranu komponentu

Što se tiče stranice *Rezervacije*, možemo definirati da je vrsti korisnika *Gosti* omogućen pregled rezervacija i njihovih plaćanja. Što se tiče interaktivne mreže *Rezervacije*, ažuriranje podataka o rezervacijama omogućeno je samo administratorima (gumb *Uredi* i *Spremi* neće biti dostupni krajnjem korisniku, a gumb *Dodaj redak* je u potpunosti ukonjen iz interaktivne mreže jer postoji forma za unos nove rezervacije – slika 93.). Kod tablice *Plaćanja* korisniku je omogućen unos novog plaćanja, no nije mu omogućeno uređivanje postojećih plaćanja (slika 94).



Slika 93. Autorizacijske postavke za interaktivnu mrežu Rezervacije



Slika 94. Autorizacijske postavke za interaktivnu mrežu Plaćanje

Nad stranicom *Nova rezervacija* nije potrebno uvoditi dodatne promjene, ona treba biti dostupna i administratorima i korisnicima aplikacije. Stranica *Korisnici* neka bude dostupna isključivo administratorima aplikacije. Dakle, u panelu s lijeve strane označimo konkretnu stranicu te joj dodijelimo autorizacijsku shemu *Administratori*. Isto učinimo i za stranicu *Korisnički računi*. Ovim dvjema stranicama moguće je pristupiti putem navigacije, no ukoliko to pokuša učiniti korisnik koji nije administrator, javlja mu se greška te mu se onemogućuje pristup stranici. Kako bismo ovo izbjegli, najbolje je ove dvije stranice učiniti nevidljivima

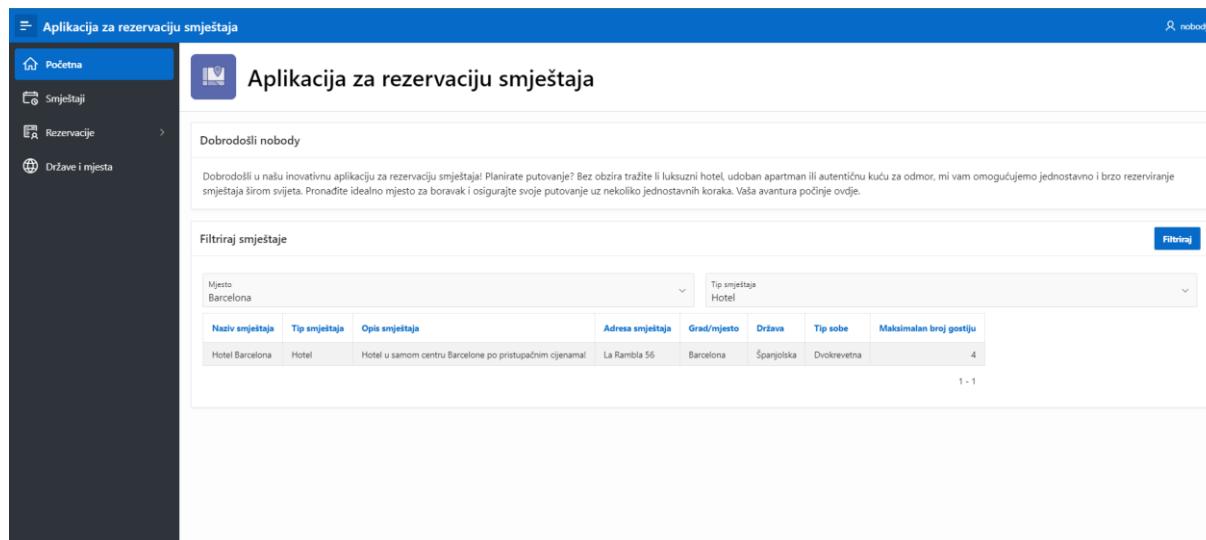
korisnicima koji nisu administratori, odnosno ukloniti ih iz navigacijske trake za sve ostale vrste korisnika. Način na koji ovo možemo učiniti opisan je ranije u tekstu.

Za svaki slučaj, stranicama *Nova rezervacija*, *Novi korisnik*, *Novi smještaj* i *Cijena* postavljamo autorizacijsku shemu *Administratori* kako bismo osigurali da ostali korisnici ipak ne bi na neki način došli do njih (iako su one uklonjene iz navigacijske trake i uklonjeni su gumbi koji vode na njih).

Još je preostalo definirati postavke za stranicu *Države i mjesta*. Neka korisnicima bude omogućen samo pregled ovih interaktivnih mreža, dok će administratorima biti omogućeno i ažuriranje, brisanje te dodavanje podataka o istima.

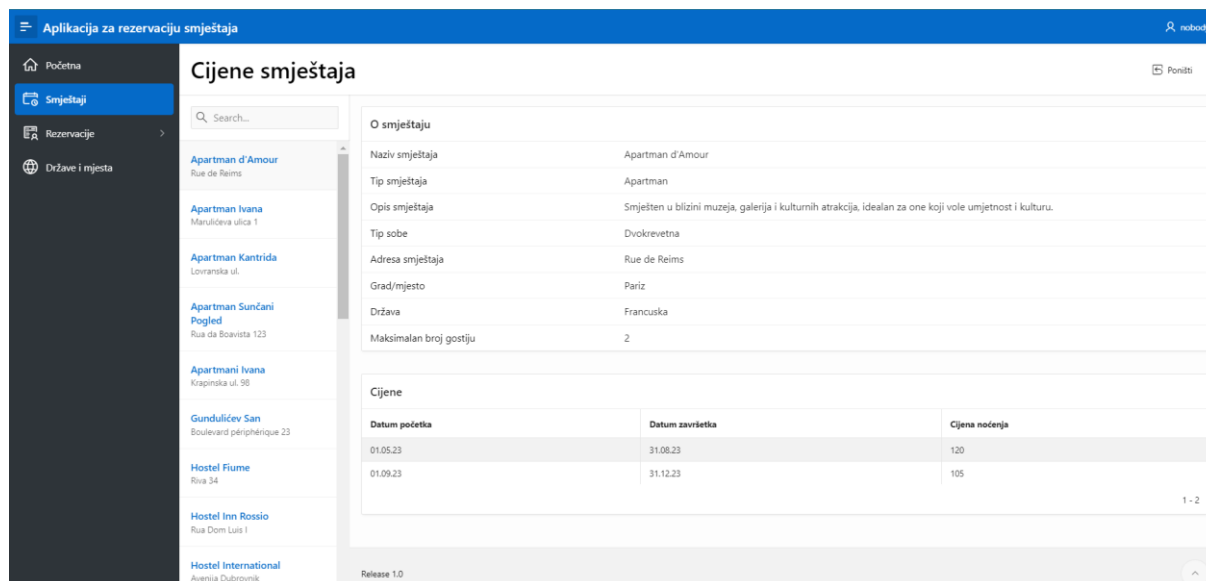
6. Prikaz kreirane aplikacije

Naslovna stranica aplikacije prikazana je slikom 95. U gornjem desnom kutu primjećujemo status prijave korisnika, koji je trenutno označen kao *nobody*, što implicira da trenutno nitko nije prijavljen u aplikaciju. Slika također jasno prikazuje primijenjeni filter koji sužava prikaz smještajnih jedinica na hotele smještene isključivo u Barceloni. Klikom na naslov pojedinog stupca dobivenog izvješća, moguće je izvršiti sortiranje podataka prema vrijednostima zapisanim u tom stupcu.



Slika 95. Početna stranica

Odabirom stranice *Smještaji*, dolazimo do ekrana prikazanog slikom 96. Ova stranica nudi popis svih smještajnih jedinica, a klikom na neku od njih dolazimo do više podataka o tome smještaju, kao i do svih cijena koje su sa njim povezane.



Slika 96. Stranica Smještaji

Ukoliko se u aplikaciju prijavimo kao administrator (isključivo s tom ulogom), na stranici *Smještaji* dodatno se nalaze i gumbi kojima je moguće unijeti novi smještaj ili urediti postojeći te za njega unijeti novu cijenu ili pak urediti postojeću (vidi sliku 97).

Aplikacija za rezervaciju smještaja admin

Cijene smještaja

Search...

O smještaju

Naziv smještaja	Apartman d'Amour
Tip smještaja	Apartman
Opis smještaja	Smješten u blizini muzeja, galerija i kulturnih atrakcija, idealan za one koji vole umjetnost i kulturu.
Tip sobe	Dvokrevetna
Adresa smještaja	Rue de Reims
Grad/mjesto	Pariz
Država	Francuska
Maksimalan broj gostiju	2

Cijene

	Datum početka	Datum završetka	Cijena noćenja
<input checked="" type="checkbox"/>	01.05.23	31.08.23	120
<input checked="" type="checkbox"/>	01.09.23	31.12.23	105

1 - 2

Release 1.0

Slika 97. Stranica Smještaji ukoliko se prijavimo kao administrator

Na stranici *Rezervacije* pružen je pregled svih rezervacija prijavljenog korisnika, na način ilustriran slikom 98. Ukoliko smo prijavljeni s ulogom administratora, regija *Rezervacije* prikazivat će popis rezervacija svih korisnika, odnosno sve rezervacije koje u danom trenutku postoje u bazi podataka. Kada korisnik odabere pojedinu rezervaciju, prikazuju se sva plaćanja koja su s njom povezana. Također, korisnicima je omogućeno dodavanje novih plaćanja za prethodno izvršene rezervacije.

Aplikacija za rezervaciju smještaja mariasv

Rezervacije

Pretraživanje: svi stupci s tekstom Idi Akcije

Datum prijave	Datum odljave	Broj noćenja	Ukupna cijena	Naziv smještaja	Korisničko ime
08.10.23	01.11.23	24	2520	Hostel Via Augusta	MARIASV
22.10.23	23.10.23	1	105	Hotel Praha Lisboa	MARIASV
15.09.23	20.09.23	5	525	Hotel Barcelona	MARIASV
09.09.23	17.09.23	8	840	Gundulićev San	MARIASV

Odabran sljedeći broj redaka: 1 Ukupno 4

Plaćanja

Pretraživanje: svi stupci s tekstom Idi Akcije Uredi Spremi Dodaj redak

Način plaćanja	Iznos plaćanja	Datum plaćanja
Gotovina	2520	17.09.23

Ukupno 1

Release 1.0

Slika 98. Stranica Rezervacije

Forma za unos nove rezervacije prikazana je slikom 99. Ovdje korisnik jednostavno unosi potrebne podatke o novoj rezervaciji te ju potvrđuje klikom na gumb *Rezerviraj*. Pritom je *Naziv smještaja* potrebno odabrati iz padajuće liste, a vrijednosti u poljima *Broj noćenja* i *Ukupna cijena* izračunavaju se automatski nakon unosa potrebnih podataka u prethodna polja. Na slici su također vidljive i poruke o pogreškama do kojih dolazi u slučaju unosa neispravnih datuma prijave i odjave.

The screenshot shows the 'Nova rezervacija' (New reservation) form. The left sidebar contains navigation links: Početna, Smještaji, Rezervacije, Nova rezervacija, and Države i mjesta. The main form area has the following fields and values:

- Naziv smještaja: Gundulićev San
- Korisničko ime: MARIASV
- Datum prijave: 29.07.23
- Datum odjave: 14.07.23
- Broj noćenja: -15
- Ukupna cijena: -1575

Below the date fields, there are two error messages in red text:

- 'Rezervacije su dostupne samo za buduće datume. Molimo Vas odaberite datum prijave koji pada nakon današnjeg datuma.'
- 'Datum odjave mora slijediti nakon datuma prijave. Molimo izaberite ispravne datume za Vašu rezervaciju.'

At the bottom right, there is a blue 'Rezerviraj' button and an 'Odustani' button on the left.

Slika 99. Stranica za unos nove rezervacije

Način prikaza država i njima pripadajućih mjesta ostvaren je prema slici 100. Na lijevoj strani zaslona omogućen je pregled svih unesenih država iz baze podataka. Klikom na pojedini redak u tom popisu, na desnoj strani zaslona prikazuju se sva mjesta smještena unutar te odabrane države, a koja su pohranjena u bazi podataka nad kojom je ova aplikacija kreirana.

The screenshot shows the 'Države i mjesta' (Countries and places) page. The left sidebar contains navigation links: Početna, Smještaji, Rezervacije, Države i mjesta, Korisnici, and Korisnički računi. The main area is divided into two panels:

- Popis država:** A table with columns for 'Naziv države'. It lists Portugal, Španjolska, Hrvatska (selected), and Francuska. At the bottom, it shows 'Odabran slijedeći broj redaka: 1' and 'Ukupno 4'.
- Popis mjesta:** A table with columns for 'Poštanski broj' and 'Naziv grada ili mjesta'. It lists 10000 (Zagreb), 23000 (Zadar), and 51000 (Rijeka). At the bottom, it shows 'Odabran slijedeći broj redaka: 1' and 'Ukupno 3'.

Both panels have a search bar and a 'Ponovo postavi' button. The 'Popis mjesta' panel also has a 'Ponovo postavi' button.

Slika 100. Stranica s popisom država i mjesta

Posljednje dvije stranice ove aplikacije nose nazive *Korisnici* (slika 101) te *Korisnički računi* (slika 102). Ovim stranicama mogu pristupiti jedino oni korisnici koji pripadaju pod autorizacijsku shemu *Administratori*, a one omogućuju upravljanje korisnicima i korisničkim računima aplikacije.

Aplikacija za rezervaciju smještaja									
Korisnici									
<div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div>									
OIB	Ime	Prezime	Adresa	Email adresa	Broj telefona	Korisničko ime	Naziv države	Naziv mjesta	
45367009843	Josip	Josipović	Josipovića 25B	josipovic@gmail.com	996789807	JOSIPIOS	Hrvatska	Rijeka	
65489320125	Mirela	Mirić	Vukovarska ul. 65	mirelam@gmail.com	956821354	MIRELAM	Hrvatska	Zagreb	
12345678910	Marko	Marković	Marulićeva 32	markom@gmail.com	951234567	MARKOM	Hrvatska	Rijeka	
8888888	Maria	Siročić Vujnović	Antuna Banca 18	maria.sv@student.unin.hr	955023164	MARIASV	Hrvatska	Rijeka	
34567890987	Anita	Move	Gundulićeva 14	movea@gmail.com	95622331	Anita M	Hrvatska	Rijeka	
96521453658	Jovica	Jović	Dubrovačka 43	jovic@gmail.com	998563214	JOVICA	Hrvatska	Zadar	
81	Perko	Perković	Radmile Matejčić 2	perkoP@gmail.com	456	PERKO	Hrvatska	Rijeka	
3265987410	Janko	Janković	La Guardia 56b	jankovic@gmail.com	38596320156	Jakovici	Španjolska	Barcelona	
86541236478	Marta	Mirić	Zvonimirova 56	miricm@gmail.com	956897532	MARTA MIRIC	Hrvatska	Zadar	
10987654321	Ana	Anić	Krešimirova 2A	anicana@gmail.com	919876543	ANAA	Hrvatska	Zagreb	
74196385201	Ivan	Ivanović	Kralja Petra Krešimira 61	ivan@gmail.com	956324812	IVAN	Hrvatska	Zadar	

Slika 101. Stranica Korisnici

Aplikacija za rezervaciju smještaja									
Korisnički računi									
Korisničko ime	Lozinka	Administrator?							
ADMIN	pk3l3ck3zkW_8V55bMq7bnjVvqj3N69Hzci1e15IgeHdP1qz7qTo7S1qV124TeeReshazUEAA	Y							
ANITA M	3N7-4uRgM1guCfMqalot_e1Vw-s35m491VAD7oSLuDEF0b8fC3d0bY_1e9p006fFp9p5nZ3qJX0h61Q	N							
GUEST	wR8denv_BH-4bb1afPHeMwVm5u36FYQorap5pU0zqUyng7ThhgmpTeRiWb4ODIM5WfwbwvO_bNvyQg	N							
IVAN	eqhMwVcR3K7Q74eIevm-CZehJlA3PeI-nE36egLe8RTwODQJNirFRdCH5Wjy3R7-4V1N0L6CB3p9g	N							
JAKOVICJ	dN4ku162Zu11ind4855DCzT1lp8E6etgCgkHCTA9SkvWKR2bNW4W0z0MqyHRTEdwaF8Y5k84CheOWBv	N							
MARIASV	-CZd8DeH9D_pMUqec5-tvvpV0LHCpZYCn5Fq20W9g0MKD_RDEgBXUaMC3wo1gnpPcE8Doa16hRmChQ	N							
MARKO M	sa4f5voUg7v8emaZY-48FvfwkGJfmdOyPhgLE_xmBbnkEIQz9ohhng1aWq4mtrB8uWuSkud4yV0pLA	N							
MIRELAM	UkCh24kq8FXG3qyOK8QJVF0z0K0yWacqDwvPQZWEhBIPqBfap4TZQ0n19qbrgghZyrlB3R1QpR0gQ	N							
PERKO	R2AJS4F-QndOFF4by4F35EMHvg4CmNupT95uu3gzvmmkGPZ7p3YKatzLfYfANWCaN1HybKuyjeXlIQa	N							

Unos novog korisnika

Korisničko ime

admin

Lozinka

Potvrdi lozinku

Je li ovaj korisnik administrator?

☐

Unesi

Slika 102. Stranica Korisnički računi

7. Zaključak

Kroz detaljan proces razvoja aplikacije za rezervaciju smještaja – od definiranja korisničkih modela do implementacije ključnih funkcionalnosti – ovaj rad naglašava nekoliko ključnih spoznaja. Prvo, Oracle APEX pruža moćan okvir za brz i efikasan razvoj web aplikacija. Drugo, modeliranje korisnika i korisničke priče omogućava usklađivanje razvoja s stvarnim potrebama korisnika, poboljšavajući njihovo iskustvo. Treće, integracija dijagrama entiteti-veze i relacijskog modela omogućila je strukturiranje baze podataka koja podržava ključne operacije rezervacije smještaja. Četvrto, izrada aplikacije koja omogućava CRUD operacije na bazi podataka pruža jasnoću i praktičnost u upravljanju rezervacijama. Konačno, stavljena je važnost na provođenje autentifikacijskih i autorizacijskih postupaka kako bi se garantirala sigurnost aplikacije, ali i njezinih podataka.

U današnjem brzom tempu života i rastućoj digitalnoj interakciji, ovakve aplikacije su ključne za olakšavanje rutinskih zadataka i poboljšanje korisničkog iskustva. Kroz ovaj rad cilj je bio naglasiti potencijal Oracle APEXA u stvaranju visoko funkcionalnih i korisnički orijentiranih poslovnih aplikacija.

U budućnosti, daljnja istraživanja mogla bi se fokusirati na optimizaciju performansi aplikacije, unapređenje sigurnosnih aspekata te širenje funkcionalnosti kako bi se obuhvatile različite tipove smještaja ili dodatne usluge. Ovaj rad ostavlja otvoren prostor za daljnje inovacije i razvoj unutar ovog dinamičnog područja digitalnih poslovnih aplikacija.

8. Popis literature

Čandrlić, S. (2023.). Materijali s predavanja kolegija *Dizajn korisničkog sučelja i iskustva* te *Uvod u programsko inženjerstvo*.

Oracle. (2023.). *What is Oracle APEX*. Dohvaćeno iz Oracle APEX: <https://apex.oracle.com/en/>, pristupljeno: kolovoz 2023.

Pavlić, M. (2011.). *Oblikovanje baza podataka*. Rijeka: Odjel za informatiku Sveučilišta u Rijeci.

Sciore, E. (2020.). *Understanding Oracle APEX 20 Application Development*. Appress, MA, USA.

9. Popis slika

Slika 1. Model prvog korisnika	8
Slika 2. Model drugog korisnika	9
Slika 3. Model trećeg korisnika.....	9
Slika 4. Koncepti modela entiteti-veze (Pavlić, 2011.)	10
Slika 5. Prikaz dijagrama entiteta-veza za aplikaciju za rezervaciju smještaja	12
Slika 6. Različiti načini korištenja Oracle APEX-a.....	17
Slika 7. Slanje zahtjeva za novo radno okruženje	18
Slika 8. E-mail za završetak kreiranja radnog okruženja	18
Slika 9. Prijava u radno okruženje.....	19
Slika 10. Početni zaslon radnog okruženja.....	19
Slika 11. Sučelje SQL Workshop alata	20
Slika 12. Sučelje alata Object Browser	21
Slika 13. Sučelje alata SQL Commands.....	21
Slika 14. Kreiranje tablice DRZAVA kroz SQL Commands alat	22
Slika 15. Prikaz tablice DRZAVA u alatu Object Browser	22
Slika 16. Odabir opcije za kreiranje nove tablice.....	23
Slika 17. Kreiranje tablice MJESTO kroz alat Object Browser	24
Slika 18. Čarobnjak za dodavanje ograničenja vanjskog ključa u tablicu MJESTO.....	25
Slika 19. Unos nove države u tablicu DRZAVA	26
Slika 20. Unos novog mjesta u tablicu MJESTO	26
Slika 21. Pogreška prilikom narušavanja ograničenja vanjskog ključa.....	27
Slika 22. Ispunjena forma za kreiranje tablice REZERVACIJA.....	27
Slika 23. Kreiranje ograničenja vanjskog ključa za stupac SIFRA_SMJESTAJA u tablici REZERVACIJA	28
Slika 24. Sučelje alata App Builder.....	29
Slika 25. Mogućnosti prilikom kreiranja nove aplikacije.....	29
Slika 26. Definiranje postavki nove aplikacije.....	30
Slika 27. Početno sučelje aplikacije	30
Slika 28. Dodavanje nove regije.....	31
Slika 29. Dodavanje nove regije putem drag and drop opcije.....	32
Slika 30. Definiranje izvora podataka za statično izvješće	32
Slika 31. Postavljanje stupca kao skrivenog.....	32
Slika 32. Sučelje alata Shared Components	33
Slika 33. Prvi zaslon čarobnjaka za kreiranje nove liste vrijednosti	34
Slika 34. Drugi zaslon čarobnjaka za kreiranje nove liste vrijednosti.....	34
Slika 35. Treći zaslon čarobnjaka za kreiranje nove liste vrijednosti.....	35
Slika 36. Četvrti zaslon čarobnjaka za kreiranje nove liste vrijednosti	36
Slika 37. Pridruživanje liste vrijednosti TIP_SMJESTAJA stupcu SIFRA_SMJESTAJA	36
Slika 38. Pridruživanje liste vrijednosti NAZIV_MJESTA padajućem izborniku P1_MJESTO_FILTER	37
Slika 39. Pridruživanje liste vrijednosti TIP_SMJESTAJA padajućem izborniku P1_TIP_SMJESTAJA_FILTER	37
Slika 40. Dodavanje novog gumba pomoću drag and drop opcije.....	37
Slika 41. Definiranje ponašanja i postavki gumba	38
Slika 42. Definiranje izvora podataka izvješća kao rezultata SQL upita	38
Slika 43. Prvi korak prilikom kreiranja Master Detail stranice	39

Slika 44. Drugi korak prilikom kreiranja Master Detail stranice	39
Slika 45. Treći korak prilikom kreiranja Master Detail stranice	40
Slika 46. Posljednji korak prilikom kreiranja Master Detail stranice.....	41
Slika 47. Kreiranje nove prazne stranice.....	42
Slika 48. Definiranje postavki stranice.....	42
Slika 49. Definiranje postavki regije Rezervacije	43
Slika 50. Dozvoljavanje operacija ažuriranja i brisanja redaka interaktivne mreže.....	43
Slika 51. Dodavanje novog stupca	44
Slika 52. Prvi dio postavki stupca NAZIV_SMJESTAJA	44
Slika 53. Drugi dio postavki stupca NAZIV_SMJESTAJA.....	45
Slika 54. Definiranje master regije za regiju Plaćanje.....	45
Slika 55. Definiranje Master stupca za stupac SIFRA_SMJESTAJA tablice PLACANJE	46
Slika 56. Definiranje Master stupca za stupac OIB tablice PLACANJE	46
Slika 57. SQL upit za kreiranje okidača "provjera_plaćanja"	46
Slika 58. Definiranje Master regije	47
Slika 59. Definiranje Master stupca	47
Slika 60. Kreiranje nove stranice kao forme	47
Slika 61. Kreiranje forme Nova rezervacija	48
Slika 62. Posljednji korak prilikom kreiranja nove forme.....	49
Slika 63. Kartica Dynamic Actions	49
Slika 64. Kreiranje nove dinamičke akcije.....	50
Slika 65. Definiranje postavki dinamičke akcije.....	50
Slika 66. Definiranje postavki za True granu akcije BrojNocenja	51
Slika 67. Definiranje postavki za True granu akcije UkupnaCijena	51
Slika 68. Kreiranje stranice interaktivnog izvješća i njoj pripadne forme za unos podataka	52
Slika 69. Odabir primarnog ključa interaktivnog izvješća	53
Slika 70. Kreiranje nove validacije	53
Slika 71. Validacija za provjeru dostupnosti smještaja s obzirom na datum prijave	54
Slika 72. Validacija za provjeru dostupnosti smještaj s obzirom na datum odjave.....	54
Slika 73. Validacija za provjeru datuma prijave u odnosu na današnji datum	55
Slika 74. Validacija za provjeru datuma odjave u odnosu na datum prijave.....	55
Slika 75. Kreiranje tablice USERS i umetanje podataka u nju	56
Slika 76. Ispis podataka iz tablice USERS	56
Slika 77. Hijerarhijsko drvo stranice Korisnički računi	56
Slika 78. Postavke za gumb Unesi unutar regije Unos novog korisnika	57
Slika 79. Validacija za provjeru unesenog korisničkog imena.....	58
Slika 80. Validacija za provjeru podudarnosti lozinki	58
Slika 81. Prvi dio postavki procesa za unos novog korisnika	59
Slika 82. Drugi dio postavki procesa za unos korisnika.....	59
Slika 83. Prvi zaslon čarobnjaka za stvaranje nove autentifikacijske sheme	59
Slika 84. Drugi zaslon čarobnjaka za stvaranje nove autentifikacijske sheme.....	60
Slika 85. Definiranje stranice kao javno dostupne	61
Slika 86. Prvi zaslon čarobnjaka za kreiranje nove autorizacijske sheme.....	61
Slika 87. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme Administratori	62
Slika 88. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme Gosti	63
Slika 89. Drugi zaslon čarobnjaka za kreiranje autorizacijske sheme Buit-in korisnici	63
Slika 90. Postavljanje autorizacije nad stranicom navigacijskog izbornika	64
Slika 91. Odabir komponente nad kojom želimo postaviti autorizacijsku shemu.....	64

Slika 92. Definiranje autorizacijske sheme za odabranu komponentu	65
Slika 93. Autorizacijske postavke za interaktivnu mrežu Rezervacije.....	65
Slika 94. Autorizacijske postavke za interaktivnu mrežu Plaćanje	65
Slika 95. Početna stranica.....	67
Slika 96. Stranica Smještaji.....	67
Slika 97. Stranica Smještaji ukoliko se prijavimo kao administrator.....	68
Slika 98. Stranica Rezervacije.....	68
Slika 99. Stranica za unos nove rezervacije	69
Slika 100. Stranica s popisom država i mjesta	69
Slika 101. Stranica Korisnici.....	70
Slika 102. Stranica Korisnički računi.....	70

10. Popis priloga

Poveznica na aplikaciju za rezervaciju smještaja:

<https://apex.oracle.com/pls/apex/f?p=128434:1>

Podaci za prijavu

- Korisničko ime: ADMIN, lozinka: 1234 (admin)
- Korisničko ime: MARIASV, lozinka: mariasv123 (guest)
- Korisničko ime: IIVAN, lozinka: ivan123456 (guest)
- Korisničko ime: JAKOVICJ, lozinka: jakovicj (guest)