Primjer aplikacije za potporu upravljanja turističkom destinacijom (DMC)

Lazarević, Beate

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:143059

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike

Beate Lazarević

PRIMJER APLIKACIJE ZA POTPORU UPRAVLJANJA TURISTIČKOM DESTINACIJOM (DMC)

Završni rad

Pula, 2019.

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike

Beate Lazarević

PRIMJER APLIKACIJE ZA POTPORU UPRAVLJANJA TURISTIČKOM DESTINACIJOM (DMC)

Završni rad

JMBAG:0303038001, redoviti student
Studijski smjer: Informatika

Predmet: Elektroničko poslovanje
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijsko-komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentorica: prof. dr. sc. Vanja Bevanda

Pula, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za magistra ekonomije/poslovne ekonomije ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA
o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom _____ _____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis
_____

SADRŽAJ:

UVOD 1
1. Destinacijski menadžment kompanija (DMC) 3
1.1 Tvrtka „Destination Touristic Services“ (DTS) 4
1.2 Upravljanje i usluge DTS-a 5
1.3 Transferi DTS-a 6
2. Proces izrade aplikacije "TRANSFERI" 7
2.1 Opis „Aplikacije“ 8
2.2 Specifikacija zahtjeva 9
2.2.1 Podjela zahtjeva prema razini detalja 9
2.2.2 Podjela zahtjeva prema sadržaju 10
3. Programski alati korišteni za izradu aplikacije 12
3.1 XAMPP 12
3.2 PhpMyAdmin 13
3.3 NetBeans 15
4. Primjena aplikacije 27
ZAKLJUČAK 32
SAŽETAK 34
POPIS LITERATURE 35
POPIS SLIKA 37
SUMMARY 38

UVOD

Završni rad se temelji na izradi aplikacije "Transferi" koja bi mogla postati značajna potpora u upravljanju turističkom destinacijom. Rad se sastoji od sljedećih poglavlja: uvoda, općenito o destinacijskom menadžmentu (DMC), razvoju projekta, procesu izrade aplikacije, programskim alatima, primjeni aplikacije, zaključka i literature.

U prvom poglavlju „Destinacijski menadžment“, opisani su glavni pojmovi turističke kompanije i njezina struktura i funkcioniranje. Predstavljena je odabrana turistička agencija "Destination Touristic Services" i turističke usluge koje pruža, a u koje se između ostalog ubrajaju i transferi. Ručni unos podataka o transferima putem Excel dokumenta pri radu stvara određene probleme. Ideja vodilja je izrada aplikacije koja će se odnositi na transfere i omogućiti transparentniju kontrolu dolazaka i odlazaka gostiju, te pojednostaviti komuniciranje s transfer-partnerima. Ponuđenim rješenjem ova aplikacija bi u velikoj mjeri doprinijela unapređenju destinacije.

U drugom i trećem poglavlju opisan je razvoj i proces izrade same aplikacije uz primjenu programskih alata kao što je NetBeans, a za izradu same baze odabran je XAMPP web server otvorenog koda i besplatni software phpMyAdmin. Opisana je i specifikacija zahtjeva, te glavne karakteristike aplikacije. Prikazan je izgled aplikacije s početnom stranicom prijave korisnika „Log in“, te izgled kod unosa i pregleda transfera.

Tema četvrtog poglavlja je primjena aplikacije na zaposlenike koji bi je svakodnevno koristili. Upotrebljivost i korisnost ove aplikacije je u jednostavnijim riješenima u organizaciji i radu na uslugama prijevoza gostiju u samoj destinaciji. S obzirom na to da turizam u Hrvatskoj iz godine u godinu bilježi stalni rast, podaci su prateći taj trend sve opsežniji, te ručni unos koji se trenutno koristi u agenciji zna biti nepregledan i vremenski zahtjevan. Ovaj projekt posjeduje veliki komercijalni potencijal, te bi se njegovom dodatnom nadogradnjom mogao implementirati u mnoge destinacije, znatno

olakšavajući i ubrzavajući sam proces organizacije transfera gostiju i popratnih usluga koje su uz njega vezane.

1. Destinacijski menadžment kompanija (DMC)

Turistička potražnja je dinamičan proces, konkurencija je sve jača, a gosti sve zahtjevniji. Zato kreatori turističke ponude nastoje odgovoriti stvaranjem novih, inovativnih i zanimljivih turističkih proizvoda. Za uspjeh na tržištu ponuda se treba prilagođavati svim zahtjevima, pri čemu važnu ulogu ima destinacija u kojoj se ponuda stvara, te njezine specifičnosti i komparativne prednosti. Nositelji destinacijskog menadžmenta su destinacijske menadžment organizacije, koje kroz svoje aktivnosti stvaraju kvalitetu proizvoda.

Pojam "Destination managment company" (DMC), odnosno prevedeno s engleskog: "Destinacijska menadžment kompanija" (DMK), nastao je početkom sedamdesetih godina u Sjedinjenim Američkim Državama, radi opisivanja sve aktivnije uloge operatora kao specijaliste za logistiku u turističkoj destinaciji. DMC možemo definirati kao profesionalno uslužno poduzeće koje raspolaže opsežnim znanjem o (geografskom) području u kojem posluje, o lokalnim događanjima i aktivnostima, te se bavi organizacijom aranžman-tura, prijevoza i programskom logistikom. [1]

U nastavku Završnog rada prikazana je jedna odabrana turistička kompanija pod nazivom "Destination Touristic Services" uz osvrt na trenutno upravljanje destinacijom, te je posebno obrađena problematika vezana uz transfere.

[1]Dr. Sc. Sandra Čorak, Željko Trenzer (2014), Destinacijski menadžment kompanije, str.14.

1.1 Tvrtka „Destination Touristic Services“ (DTS)

Tvrtka „Destination Touristic Services“ (DTS) je međunarodna destinacijska menadžment kompanija za pružanje kompleksnih turističkih usluga. Kao globalna agencija receptivnog tipa dio je turističke grupacije, turističkog sektora „Rewe“ koji posluje pod nazivom „DER Touristik“ i po veličini je treći organizator putovanja u Europi. DTS nudi široku paletu visokokvalitetnih usluga za organizatore putovanja na različitim tržištima. [2]

DTS s 34 ureda pokriva najtraženija turistička odredišta u 10 zemalja na 3 kontinenta. Svaki ured ima tim stručnjaka koji osiguravaju jedinstvenu uslugu i jamče nesmetani rad svim turističkim agencijama. Kako bi se nadgledali svi uredi na odredištima, formirano je neovisno sjedište u Palma de Mallorci, DER Touristik Services S.L.U. koje djeluje kao holding za sve podružnice koje nude turističke usluge u odredištu. Počevši s radom u studenom 2013. godine s odredišnim uredima u Egiptu, UAE, Bugarskoj i Tunisu, tvrtka neprestano raste. Otvaranjem dolaznih podružnica u Španjolskoj i Turskoj u studenom 2014. godine, odredišne turističke usluge postale su vodeće i na tržištu u Egiptu i Tunisu za njemačke turističke kuće. Od siječnja 2015. tvrtka nudi svoje usluge u Grčkoj, a od svibnja 2016., također u srednjoj Europi, sa sjedištem u Austriji i lokalnim uredima u Hrvatskoj. U 2017. tvrtka je počela nuditi svoje usluge i u Dominikanskoj republici. [3]

[2] Destination touristic services; Dostupno na: https://dtservices.travel/destinations/
[3] Destination touristic services, Destinations; Dostupno na: https://dtservices.travel/destinations/

1.2 Upravljanje i usluge DTS-a

Za upravljanje destinacijom odgovorne su organizacije koje su zadužene za operativne zadatke. Organizacije djeluju kao turistički informacijski centri koji su orijentirani isključivo na pružanje usluga posjetiteljima. Ključna zadaća nositelja upravljanja u turističkoj destinaciji je prepoznati i okupiti zaposlenike, te im osigurati alate za zajednički rad s ciljem transformacije skupa atrakcija, aktivnosti i usluga. Navedeni proces započinje planiranjem, odnosno izradom strateških, razvojnih i operativnih planova na koje se nadovezuje izrada marketinškog plana kao zasebne strategije u plasiranju proizvoda na tržište.

Globalna turistička agencija pruža širok spektar usluga visoke kvalitete turističkim operaterima na različitim tržištima. Pored klasičnih usluga smještaja, izleta i transfera, spektar usluga obogaćen je i organiziranjem grupnih putovanja, događanja i usluga vezanih uz letove do odabranih destinacija. Zahvaljujući predanom, profesionalnom i kreativnom timu, destinacijske turističke usluge rado odabiru gosti s visokim zahtjevima.

Turističke usluge tvrtke DTS-a:

- ugovori
- rent-a-car
- ture i izleti
- rezervacija hotela
- transferi
- vjenčanja i medeni mjesec
- obiteljski odmor
- predstavništvo
- turistički vodiči
- posebne usluge[4]

[4]Destination touristic services, Incoming services; Dostupno na: https://dtservices.travel/incoming-services/

1.3 Transferi DTS-a

Prema tipu vozila, usluge prijevoza dijele se na: taxi prijevoz, prijevoz limuzinom ili autobusni prijevoz. Kako bi se udovoljilo različitim zahtjevima gostiju, transferi se organiziraju kao privatni ili zajednički. Na zahtjev se može organizirati i prijevoz posebne prtljage kao što je npr. sportska oprema (oprema za golf, biciklizam, surfanje i sl.). U svrhu pružanja kvalitetne usluge transfera, partneri u destinaciji stalno provode sigurnosne provjere kako bi se eliminirale pogreške.

Sustav za transfere trenutno funkcionira na način da zaposlenici DTS-a iz sustava izvlače određenu količinu podataka, te ih ručno unose u velike Excel dokumente. Budući da se DERTouristik sastoji od više brendova, podijeljen je na dva sustava. Postoje dva trenutno aktualna programa iz kojih se izvlače liste rezervacija: „Phoenix Unlimited“ i „o.p.i. online partner interface“. Nakon izvlačenja potrebne količine podataka i informacija iz liste, posebno se sortiraju rezervacije koje sadrže transfere, te se u tablice unosi rezervacijski broj, ime, datum dolaska i odlaska, vrijeme leta i smještajna jedinica. Nakon toga tablice se kreiraju po danu, zasebno za dolaske i odlaske, te se šalju transfer-partnerima, kako bi se pravovremeno organizirao prijevoz za klijente.

Nakon slijetanja u zračnu luku, goste dočekuju turoperatori koji također dobivaju liste za tekući dan. Na turoperatoru je da animira goste dok čekaju, te ih uputi prema usluzi prijevoza. Potom goste preuzimaju transfer-partneri, koji ih u najkraćem roku prevoze do odredišta. Korisnicima privatnih transfera se pridaje poseban značaj, obzirom da za ovu uslugu plaćaju gotovo dupli iznos. Tijekom boravka u destinaciji klijenti također mogu otkazati uslugu, ali su o tome dužni pravovremeno obavijestiti turističku agenciju kako bi se usluga mogla otkazati, odnosno stornirati.

21;EW ;EW 8004;03.10.19;TXL ;PUY ;15:15;21:05
32;ITS;14665557;; ;;;;MUELLER, BIRGIT ;;38;FRAU
32;ITS;14665557;; ;;;;MUELLER, LEA MARIA ;;18;FRAU

21;EW ;EW 8962;06.10.19;TXL ;PUY ;06:05;07:40

32;ITS;14680312;; ;;;;BRETTSCHNEIDER, GOR ;;43;HERR
32;ITS;14680312;; ;;;;BRETTSCHNEIDER, DIT ;;42;FRAU
32;ITS;14680312;; ;;;;BRETTSCHNEIDER, LAU ;;15;KIND
32;ITS;14680312;; ;;;;BRETTSCHNEIDER, LEO ;;12;KIND
32;ITS;14680312;; ;;;;BRETTSCHNEIDER, LIL ;; 6;KIND

Prikaz - primjer liste rezervacija transfera iz „o.p.i.-a“

Izvor: https://opi.dertouristik.com/

2. Proces izrade aplikacije "TRANSFERI"

Jedna od ključnih zadaća upravljanja u turističkoj agenciji je okupiti kvalitetne djelatnike, te im osigurati potrebne alate za zajednički rad. Izrada desktop aplikacije pomogla bi upravljanju turističke agencije. Cilj je operativnim mjerama doći do lakšeg planiranja i organizacije.

U nastavku je prikazan postupak izrade same aplikacije za potporu upravljanja turističkom agencijom koja se odnosi na transfere. Ideja je stvoriti aplikaciju koji bi osigurala informativne sadržaje za turoperatore, tj. zaposlenike same turističke agencije. Aplikacija bi znatno pomogla u rješavanju zahtjevnog posla u kraćem vremenskom roku. Umjesto kreiranja tablica za svaki dan posebno, već bi se u samoj aplikaciji nakon unosa sortirale po danu rezervacije za koje je potrebna usluga transfera. Aplikacija je strukturirana jednostavno kako bi korisnicima osigurala kvalitetno korisničko iskustvo.

Za razvoj baze podataka korišten je XAMPP web server otvorenog koda, koji se sastoji od MYSQL baze i Apache web servera. Sama baza je izrađena u besplatnom programu u phpMyAdmin. Za dizajn i strukturu aplikacije odabran je NetBeans, koji omogućava razvoj aplikacija u mnoštvu programskih jezika, ali je primarno namijenjen Java programskom jeziku. Program je razvijen za operativne sustave Windows, Linux i Mac OS.

2.1 Opis Aplikacije

Aplikacija „Transferi“ postavljena je na način da u svakom trenutku osigurava prikaz dolazaka i odlazaka za određeni dan koji nas zanima ili nam je promptno potreban na određeni dan. Pruža uvid u sljedeće podatke: ime gosta, koliko je gostiju na rezervaciji, te na kojoj lokaciji su točno smješteni te vrijeme leta. Po dolasku u zračnu luku, turoperator raspolaže preciznim informacijama, te goste upućuje prema odabranoj soluciji prijevoza.

Transfer partneri bi također imali direktan pristup aplikaciji, što bi uveliko ubrzalo proces organizacije. Usluga transfera bi se na ovaj način mogla jednostavnije organizirati na vrijeme, vozače pripremiti za ture koje su planirane sljedeći dan, te ne bi kao do sada morali čekati na ažuriranu Excel tablicu.

Ovakvim načinom planiranja minimalizirale bi se neke od pogrešaka koje se povremeno događaju, jer bi svi partneri u transferu imali točan ispis podataka kojima raspolaže i sama agencija. Osim toga, aplikacija pruža uvid u kojim dijelovima tjedna je intenzitet pojačan, kako bi se sukladno tome prilagodio način poslovanja. Korisnici mogu prethodnog dana promijeniti podatke, ali su zbog mogućih izmjena dužni kontrolirati trenutno stanje kako bi se eliminirale eventualne pogreške.

SNAGE	SLABOSTI
-motiviranost osoba -lakši pristup informacijama -troškovne prednosti	-preopterećenost -dodavanje lažnih informacija
PRILIKE	PRIJETNJE
-velika potražnja za turizmom -mala konkurentnost -veliki potencijal nadogradnje aplikacije	-spori rast tržišta -opasnost od buduće konkurencije

Tablica 1. Prikaz SWOT analize

Swot analizom, prikaz tablica 1. možemo vidjeti ispravan odabir strategije i analizu situacije. Poduzeće treba kontinuirano sagledavati i vanjske i unutarnje čimbenike, kako bi spoznalo najbolji način za ostvarivanje željenog cilja. Zbog dinamičnih promjena na tržištu treba pratiti trenutnu situaciju: snage i slabosti predstavljaju sadašnjost, dok su prilike i prijetnje budućnost.

2.2 Specifikacija zahtjeva

Početna faza softverskog procesa, gdje se analiziraju zahtjevi na budući sustav je specifikacija zahtjeva. Rezultat je dokument o zahtjevima koji opisuje što sustav treba raditi. Specifikacijom zahtjeva bavi se tim sastavljen od razvijača softvera i budućih korisnika. Mnogi projekti zapadnu u probleme kad se razine zahtjeva međusobno miješaju, zbog toga je vrlo važno odrediti zahtjeve prema razini detalja i prema sadržaju.[5]

2.2.1 Podjela zahtjeva prema razini detalja

Zahtjeve prema razini detalja dijelimo na:

- Korisničke zahtjeve (eng. user requirements)
- Zahtjeve sustava (eng. system requirements)
- Specifikaciju programske potpore (eng. software specification)

[5] Manger, R. i Mauher, M. (2011.) Programsko inženjerstvo. Zagreb:Algebra d.o.o, str.16

Korisničke zahtjeve najčešće zadaje korisnik i to uglavnom u neformalnom, tj. nestrukturiranom obliku, a pišu se u prirodnom jeziku i crtaju jednostavnim grafičkim dijagramima. No često iz njih nije jasno kako će sustav točno funkcionirati, niti koji će biti njegovi dijelovi. [6]

Korisnički zahtjevi za izradu aplikacije „TRANSFERI“:
Kako bi korisnik pristupio aplikaciji mora posjedovati korisničko ime i lozinku. Korisniku treba biti omogućen pregled i unos podataka. Sama konstrukcija korisničkog sučelja u aplikaciji ograničiti će mogućnost pogreške. Aplikacija će omogućiti filtriranje prikaza podataka po danima i rezervacijskom broju, kako bi se olakšalo snalaženje među istima. Kako će je koristiti samo zaposlenici, odlučeno je izraditi desktop aplikaciju s lokalnom bazom.

Zahtjevi sustava predstavljaju detaljnu specifikaciju o funkcionalnosti i ograničenjima programske potpore, a pišu se strukturiranim prirodnim jezikom, posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom. Cilj definiranja zahtjeva je razumijevanje problema, potreba i želja korisnika. Zahtjev je okrenut prvenstveno prema korisnicima, te problemima, a ne prema riješenjima i implementaciji. Iz navedenog se može zaključiti da zahtjev označava kakvo ponašanje korisnik želi, bez izražavanja kako će se ponašanje ostvariti.

2.2.2 Podjela zahtjeva prema sadržaju

Prema sadržaju, zahtjevi se mogu podijeliti na:

- Funkcionalne zahtjeve (eng. functional requirements)
- Nefunkcionalne ili ostale zahtjeve (eng. non-functional, other requirements)[7]

Funkcionalni zahtjevi opisuju usluge koje bi sustav trebao pružati, te kako bi trebao reagirati i ponašati se u određenim situacijama.

[6] Jović, A., Horvat, M. I Ivošević, D (2015.) Procesi programskog inženjerstva: Oblikovanje programske potpore, Zagreb, str. 20.
[7] Jović, A., Horvat, M. I Ivošević, D (2015.) Procesi programskog inženjerstva: Oblikovanje programske potpore, Zagreb str. 21.- 22.

Funkcionalni zahtjevi za izradu aplikacije „TRANSFERI“:
Korisnik isključivo unosom podataka o korisničkom imenu može pristupiti aplikaciji te u svakom trenutku ima mogućnost unosa, pregleda i brisanje termina u bazi podataka. Pri unosu termina potrebno je unijeti sve podatke, kako bi unos bio uspješan. Korisnik mora unijeti datum, vrijeme leta, ime i prezime, smještajnu jedinicu i broj mobitela.

Nefunkcionalni zahtjevi izražavaju ograničenja vezana uz funkcije sustava, tj usluge i funkcije programskog proizvoda. Nefunkcionalni zahtjevi češće se odnose na sustav kao cjelinu nego na individualnu funkciju. Oni opisuju karakteristike koje softver mora imati, te opisuju karakteristike koje sustav postavlja u odnosu na aktivnosti i funkcije koje sustav obavlja (primjerice vremenska ograničenja, ograničenja u razvojnom procesu i sl.).[8]

Nefunkcionalni zahtjevi za izradu aplikacije „TRANSFERI“:
Aplikacija će biti razvijena kao desktop aplikacija s lokalnom bazom podataka, te će biti dostupna samo na računalu na kojem je instalirana. Teži jednostavnosti i preglednosti u izgledu svih ekrana, kako bi bila što jednostavnija za korištenje samim korisnicima. Administrator sustava je zadužen za unos korisničkog imena i lozinke u bazu kako bi omogućio rad novim korisnicima. Administrator je zadužen za unos i brisanje na bazi podataka ovisno o potrebama poslovanja.

[8] Jović, A., Horvat, M. I Ivošević, D (2015.) Procesi programskog inženjerstva: Oblikovanje programske potpore, Zagreb str. 21.- 22

3. Programski alati korišteni za izradu aplikacije

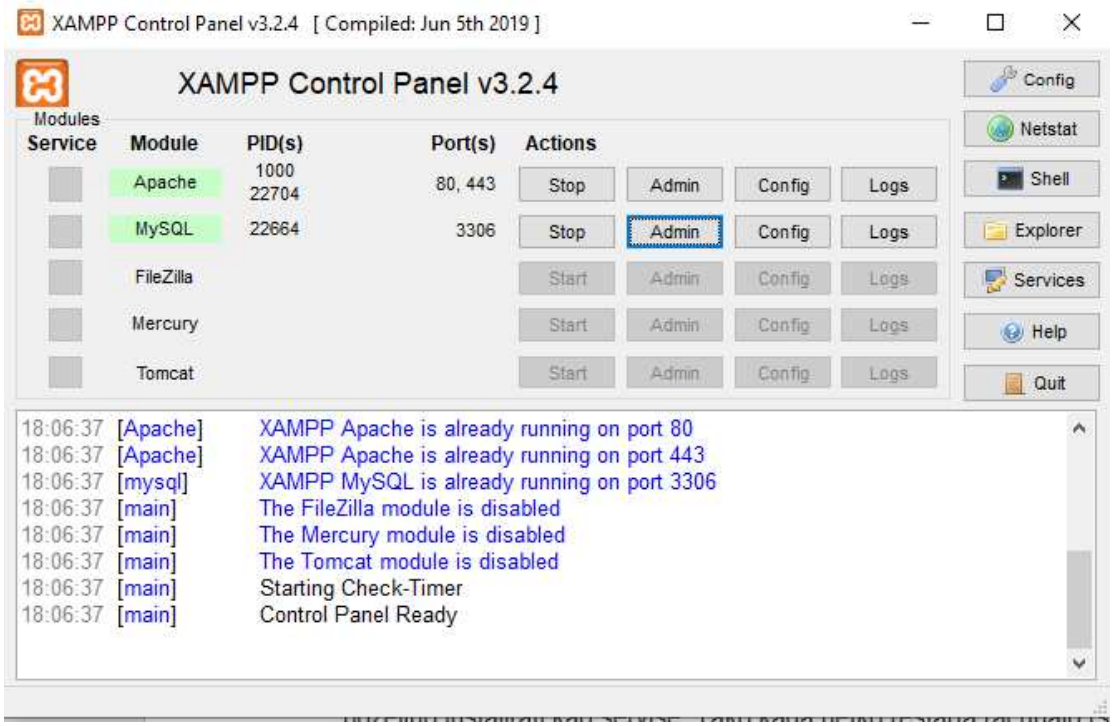
Zadatak ovog završnog rada je izrada Java desktop aplikacije za pregled i vođenje transfera. Aplikacija ima mogućnost pregleda, unosa, brisanja i izmjenu podataka koji se nalaze u MySQL bazi podataka. Detaljni prikaz i objašnjenje spajanja aplikacije s bazom podataka slijedi u narednom poglavlju.

3.1 XAMPP

XAMPP je instalacijski paket koji sadrži Apache, MySQL, PHP, phpMyAdmin i još mnoge druge aplikacije korisne za razvoj/testiranje aplikacija ili dinamičkih web stranica. Nakon što se preuzme instalacijski paket XAMPP sa službenih web stranica, pokrene se izvršna datoteka.[9]

Ovako instalirani Apache i MySQL se kao servisi automatski pokreću kada se uključi računalo.

[9]XAMPP instalacija Apache web servera za MySQL, Dostupno na; https://www.hdonweb.com/programiranje/xampp-instalacija-apache-web-servera-za-mysql-i-php



Slika 1. XAMPP Control Panel

Izvor: Autorski rad

Na slici 1. je prikazan Apache i MySQL koji su instalirani i aktivni, s lijeve strane se može vidjeti kako zapravo upravlja i da li će raditi kao servisi ili će se morati svaki puta pokretati kada bude bilo potrebe. Datotekama pristupamo putem Internet preglednika.

3.2 PhpMyAdmin

Besplatni alat za kreiranje MySQL baze pisan u PHP-u je phpMyAdmin, a služi za upravljanje i administraciju MySQL-a preko World Wide Web-a. Uz pomoć njega mogu se izvršavati mnoge MySQL operacije putem korisničkog sučelja izravno u Internet pregledniku. Možemo upravljati bazama podataka, tablicama, poljima, indeksima, korisnicima, dozvolama pristupa samim bazama, izvršavati vlastite upite na bazu i još mnogo drugih operacija.[10]

Za početak moramo imati pristup nekom MySQL serveru zajedno sa phpMyAdmin aplikacijom. Nakon što kreiramo račun i pristupimo istom, možemo kreirati svoju vlastitu bazu. Baza podataka korištena u Java desktop

[10] PhpMyadmin, About; Dostupno na: https://www.phpmyadmin.net/

aplikaciji za pregled i vođenje transfera sastoji se od dvije tablice, svaka s jedinstvenim primarnim ključem te pripadajućim redovima i stupcima.[11]

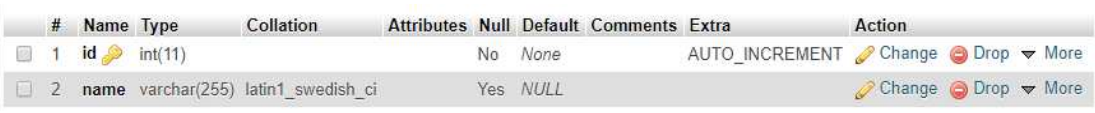
Browse | Structure | SQL | Search | Insert | Export | Import | Operations

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Uniq
2	datum_polaska	varchar(11)	utf8_general_ci		No	None		Change Drop Primary Uniq
3	vrijeme_polaska	varchar(5)	utf8_general_ci		No	None		Change Drop Primary Uniq
4	hotel	varchar(64)	utf8_general_ci		No	None		Change Drop Primary Uniq
5	broj_ljudi	int(11)			No	None		Change Drop Primary Uniq
6	ime_klijenta	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Uniq
7	kontakt_klijenta	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Uniq
8	AvioKompanijeId	int(11)			No	None		Change Drop Primary Uniq
9	vrijeme_dolaska	varchar(5)	utf8_general_ci		No	None		Change Drop Primary Uniq
10	datum_dolaska	varchar(11)	utf8_general_ci		No	None		Change Drop Primary Uniq

Slika 2.- Prikaz tablice iz baze „Rezervacije“

Izvor: Autorski rad

Baza podataka sastoji se od dvije tablice pod nazivom „Rezervacije“ i „AvioKompanije“. Slika pod brojem 1. prikazuje nam tablicu „Rezervacije“ sa svim komponentama. Svaka tablica u bazi podataka sastoji se od nekoliko elemenata/stupaca s pripadajućim tipom podataka. Možemo vidjeti da se tablica sastoji od deset stupaca. Prvi stupac je „id“ koji je ujedno i primarni ključ, zatim imamo drugi stupac „datum_polaska“, treći „vrijeme polaska“, četvrti „hotel“, peti „broj_ljudi“, šesti „ime_klijenta“, sedmi „kontakt klijenta“, osmi „AvioKompanija“, deveti „vrijeme_dolaska“, te posljednji stupac „datum_dolaska“.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More

Slika 3.- Prikaz tablice iz baze „AvioKompanije“

Izvor: Autorski rad

Na slici 3. prikazana je tablica „aviokompanija“ koja se sastoji od dva stupca. Prvi stupac je primarni ključ „id, dok je drugi stupac „name“.

[11] SQL Tutorial, dostupno na:https://www.sqltutorial.net/

3.3 NetBeans

NetBeans je integrirano razvojno okruženje tj, projekt otvorenog koda za pružanje čvrstih softverskih proizvoda za razvoj softvera.[12] Iako je NetBeans uglavnom okrenut prema Java programskim jezicima, posjeduje module za razvoj Jave, PHP, C, C++, Ruby, HTML, CSS, JavaScript i mnoge druge. Razvoj NetBeans započeo je kao studentski projekt u Češkoj, koji se prvotno zvao „Xelfi". Cilj je bio pisanje Java-IDE-a (Integrated Development Environment) nalik Delphi-u na Javi.[13]

Ono što NetBeans čini osobito zanimljivim je set RAD (Rapid Application Development) integriranih alata za izgradnju i razvoj grafičkih sučelja baziranih na standardnim AWT (Abstract Windowing Toolkit) i JFC/Swing Java Foundation Classes komponentama.[14] Potpuno je besplatan, a na svim je platformama od Windowsa do Linuxa, Solarisa i OSX-a.[15]

Za postavljanje projekta i instaliranje NetBeansa najprije moramo preuzeti Standard Edition razvojnog kompleta Java SE Development Kit (JDK) sa Java platforme. JDK je razvojno okruženje za izgradnju aplikacije i komponenti pomoću Java programskog jezika. Za ovaj projekt instaliran je JDK8 „jdk-13.0.1_windows-x64_bin". Potom je instaliran NetBeans IDE 8.2. za navedeni sustav. Točne upute mogu se potražiti na njihovim službenim stranicama.

Projekt je pisan objektno orijentiranim Java programskim jezikom. Java kodovi najprije se sastavljaju u skupu uputa za bajt kod i binarnom formatu

---

[12] Netbeans, Dostupno na: https://netbeans.org/about/

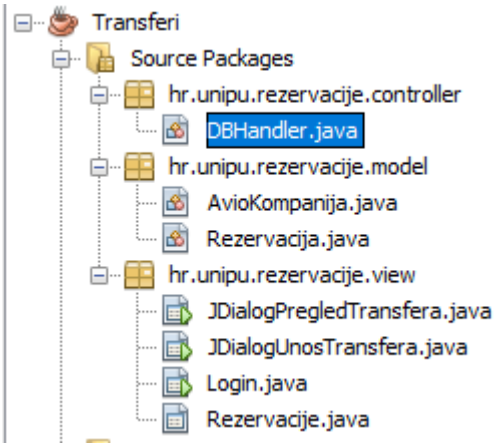
[13] History, Dostupno na: https://netbeans.org/about/history.html

[14] NetBeans, Dostupno: https://netbeans.org/features/index.html

[15] Netbeans razvojna okolin, Dostupno na: https://darko-novak.from.hr/netbeans-razvojna-okolina/

definiranom u Java virtualnom stroju JVM (Java Virtual Machine)[16]. Java sintaksa je slična C / C++, ali ona ne nudi programske funkcije niske razine poput pokazivača. Također, Java kodovi uvijek se pišu u obliku klasa i objekata. Java se koristi u svim vrstama aplikacija poput mobilnih aplikacija (Android se temelji na Javi), stolnih aplikacija, web aplikacija, klijentskih poslužiteljskih aplikacija, poslovnih aplikacija i mnogih drugih.[17]

U glavnom izborniku moramo odabrati „New project“ prevedeno znači novi projekt. Nakon toga izaberemo JavaFX kategoriju te tip projekta. Projekt je spremljen pod nazivom „Transferi“. Potrebno je novokreiranu aplikaciju oblikovati tako da radi ono što programer želi i povezati je s bazom. Potom završiti mapiranje podataka, te kreirati klase u Javi koje će omogućiti manipulaciju s podacima u bazi.



Slika 4.- Prikaz projekta “Transferi”

Izvor: Autorski rad

Prikaz slike 4. nam pokazuje arhitekturu aplikacije podjelom na tri „Java Package“ tj. paketa. Grupiranja koja su kreirana u projektu služe radi lakšeg snalaženja unutar programskog koda. Tipična aplikacija uključuje jednu ili više klasa, gdje se svaka može povezati s komponentama korištenjem na određenim stranicama. Budući da je aplikacija temeljena na „Model-View-Controller“ arhitekturi, također su i datoteke podijeljene na taj način. Unutar

[16] Java Programming Language, Dostupno na: https://docs.oracle.com/javase/7/docs/technotes/guides/language/
[17] Java, Dostupno na:https://www.geeksforgeeks.org/java/

datoteke „controller“ nalaze se Java klase koje komuniciraju s bazom podataka. Datoteka „model“ sadrži Java klase s parametrima koji odgovaraju tablicama u bazi podataka, a datoteka „view“ predstavlja grafičko korisničko sučelje vidljivo krajnjem korisniku. Prvi paket nazvan je „hr.unipu.rezervacije.controller“ i u njemu se nalazi klasa DBHHandler.java. U klasi se nalaze glavne komponente za sami rad aplikacije. Programski kod napisan unutar klase izvršava liniju po liniju te ga ispisuje u izlaznom dijelu korištenog razvojnog okruženja. Zatim u drugom paketu kreirana je klasa „hr.unipu.rezervacije.model“ koja služi za ispis aplikacije. Potom treći paket „hr.unipu.rezervacije.view“ koji sadrži četiri pogleda tj. izgled aplikacije na koji ćemo u sljedećem poglavlju detaljnije obratiti pozornost.

```
package hr.unipu.rezervacije.controller;

import hr.unipu.rezervacije.model.AvioKompanija;
import hr.unipu.rezervacije.model.Rezervacija;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.logging.Level;
import java.util.logging.Logger;
```

Slika 5.- Prikaz “Import”

Izvor: Autorski rad

Prema slici 5. vidljiv je početak sadržaja paketa „hr.unipu.rezervacije.controller“ koji nam prikazuje „import“ što znači unos koji se koristi na početku izvorne datoteke i koja može odrediti klase ili čitave pakete na koje će se kasnije pozivati bez uključivanja imena svojih paketa u referencu.[18]

[18] Import, Dostupno na: https://docs.oracle.com/javase/tutorial/information/glossary.html#import

```java
public class DBHandler {

    private final static String DB_URL = "jdbc:mysql://localhost:3306/transferi";
    private final static String USER = "korisnik";
    private final static String PASSWORD = "12345";

    public static boolean unesiRezervacijuUBazu(Rezervacija rezervacija) {
        try {
            // učitavanje drivera
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            // konekcijski objekt s parametrima za konekciju
            Connection conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
            // pripremanje upita
            String akName = rezervacija.getAvioKompanija().getName();
            AvioKompanija ak = dohvatAvioKompanije(akName);
            if(ak == null){
                PreparedStatement stmt2 = conn.prepareStatement("INSERT INTO AvioKompanije (name) VALUES (?);");
                stmt2.setString(1, akName);
                stmt2.execute();
                ak = dohvatAvioKompanije(akName);
            }

            PreparedStatement stmt = conn.prepareStatement("INSERT INTO rezervacije "
                    + "VALUES (?,?,?,?,?,?,?,?,?,?);");
            stmt.setInt(1, 0);
            stmt.setString(2, rezervacija.getDatumPolaska());
            stmt.setString(3, rezervacija.getVrijemePolaska());
            stmt.setString(4, rezervacija.getHotel());
            stmt.setInt(5, rezervacija.getBrojLjudi());
            stmt.setString(6, rezervacija.getImeKlijenta());
            stmt.setString(7, rezervacija.getKontaktKlijenta());
            stmt.setInt(8, ak.getId());
            stmt.setString(9, rezervacija.getVrijemeDolaska());
            stmt.setString(10, rezervacija.getDatumDolaska());

            stmt.execute();
            // zatvaranje konekcije
            conn.close();

        } catch (Exception ex) {
            Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
            return false;
        }
        return true;
```

Slika 6.- Prikaz klase “DBHandler”- funkcija „unesiRezervacijuUBazu“

Izvor: Autorski rad

Potom dolazimo do zahtjevnijeg dijela, odnosno pisanje programskog koda Java jezikom. Slika 6. nam prikazuje klasu DBHandler, točnije naredbe za spajanje na bazu, konekcijske objekte s parametrima za konekciju, zatim pripremamo za upit. Prikazana je jedna funkcija „public static boolean unesiRezervacijuuBazu“. „Public“ je inače metoda koja označava da je javna, tj. vidljiva je ostalim klasama, „static“ znači da je metoda konkretno povezana s određenom klasom. Za ostvarivanje veze s bazom podataka potrebno je povezati se s metodom „forName“. Nadalje otvaranje veze s bazom ostvaruje se s povezivanjem „DriverManager“ klase i njegove metode „getConnection()“ koja sadrži parametre „DB_URL,USER, PASSWORD“. Nakon uspostavljene veze koristi se SQL naredba „INSERT“ za umetanje zapisa u SQL tablicu. Prije slanja upita na bazu potrebno je kreirati objekt koji može izvršavati SQL

upit. U ovoj aplikaciji korištena je klasa „PreparedStatement“ za izradu spomenutog objekta koja prevodi SQL izraz prije izvršavanja. Parametre za SQL pruža metoda, u ovom slučaju „setString“. Nakon provjerenih parametara metoda „getOperater“ sprema zapise po id-u točno na mjesto u tablici. Varijabla „stm2“ unosi u tablicu „AvioKompanije“ a „stm“ unos u tablicu rezervacije. Kod „stmt.setInt (1,0)“ pod upitnikom „?“ nam označava mjesto gdje će se ona smjestiti. Prikazana nam je naredba „Catch“ koja služi za dohvaćanje potencijalnih grešaka koje se mogu dogoditi prilikom spajanja na bazu. Bez naredbe „Catch“ u slučaju nepovezivanja aplikacije na bazu nećemo znati zbog čega je greška nastala.

```
public static AvioKompanija dohvatAvioKompanije(String name){
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        // konekcijski objekt s parametrima za konekciju
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
        PreparedStatement stmt = conn.prepareStatement("SELECT * FROM AvioKompanije where name = ?")
        stmt.setString(1, name);
        // konekcija na bazu i slanje upita
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            AvioKompanija ak = new AvioKompanija();
            ak.setId(rs.getInt("id"));
            ak.setName(rs.getString("name"));
            return ak;
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    }
    return null;
```

Slika 7.- Prikaz klase “DBHandler”- funkcija „dohvatAvioKompanije“

Izvor: Autorski rad

U klasi „DBHandler“ nalaze se četiri funkcije. Prva je prikazana na slici 6. potom je na slici 7. prikazana funkcija „dohvatAvioKompanije“. Može se primijetiti da se naredbe za spajanje na bazu ponavljaju. Za ovu funkciju se koristi „while“ petlja koja uzastopno ponavlja blok izraza, dok je zadani uvjet istinit. Kod varijabli „ak.setId(rs.getInt("id"));“ u zagradama pod navodnicima označava se dio stupca iz baze podataka, točnije stupac iz tablice, koji se

koristi za ispis. Za obradu petlje dohvaćenih podataka koristi se objekt „ResultSet“ gdje parametar „rs“ predstavlja kursor.[19]

```
public static List<Rezervacija> dohvatRezervacijeIzBaze(String datumPolaska, String vrijemePolaska, String datumDolaska, String vrijemeDolaska, String klijent) {
    List<Rezervacija> listaRezervacija = new ArrayList<>();
    try {
        // učitavanje drivera
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        // konekcijski objekt s parametrima za konekciju
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
        PreparedStatement stmt = conn.prepareStatement("SELECT a.*, b.name as AvioKompanija FROM rezervacije as a LEFT JOIN AvioKompanije as b ON b.id = a.AvioKompanijeId;");
        // konekcija na bazu i slanje upita
        ResultSet rs = stmt.executeQuery();
        // odgovor i prikaz
        while (rs.next()) {
            Rezervacija rezervacija = new Rezervacija();
            rezervacija.setId(rs.getInt("id"));
            rezervacija.setDatumPolaska(rs.getString("datum_polaska"));
            rezervacija.setVrijemePolaska(rs.getString("vrijeme_polaska"));
            rezervacija.setDatumDolaska(rs.getString("datum_dolaska"));
            rezervacija.setVrijemeDolaska(rs.getString("vrijeme_dolaska"));
            rezervacija.setHotel(rs.getString("hotel"));
            rezervacija.setBrojLjudi(rs.getInt("broj_ljudi"));
            rezervacija.setImeKlijenta(rs.getString("ime_klijenta"));
            rezervacija.setKontaktKlijenta(rs.getString("kontakt_klijenta"));

            AvioKompanija aviokompanija = new AvioKompanija();
            aviokompanija.setName(rs.getString("AvioKompanija"));
            rezervacija.setAvioKompanija(aviokompanija);

            listaRezervacija.add(rezervacija);
        }
```

Slika 8.- Prikaz klase “DBHandler”- funkcija „dohvatRezervacijeIzBaze“

Izvor: Autorski rad

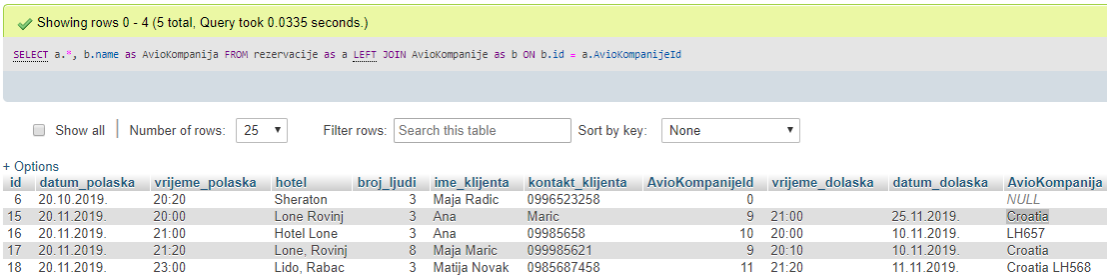
Kod treće funkcije „dohvatRezervacijeIzBaze“ imamo upit „SELECT“ koji nam je prikazan na slici 8. "SELECT a.*, b.name AS AvioKompanija FROM rezervacije AS a LEFT JOIN [20] AvioKompanije AS b ON b.id = a.AvioKompanijeId" je upit za dohvaćanje svih podataka rezervacija tako da se umjesto „AvioKompanijeID“ iz tablice rezervacije, prikaže naziv „AvioKompanije“ odgovarajućeg ID-a iz tablice „AvioKompanije“. Prvi dio "SELECT a.*, b.name as AvioKompanija" odnosi se na stupce koje želimo dohvatiti iz odgovarajućih tablica: - a.* = sve stupce iz tablice 'a' (znak * označava da se žele dohvatiti SVI stupci iz neke tablice)- b.name AS AvioKompanija = samo stupac 'name' iz tablice 'b', te postavljamo novi naziv dohvaćenog stupca (eng. alias) kao 'AvioKompanija'.

Alias je naziv za dohvaćeni stupac ili tablicu putem „query-a“, kojeg mu mi implicitno zadamo, dok originalan naziv stupca se u bazi podataka ne mijenja. Alias se može koristiti već u samom „queryu“ tako da neki stupac možemo referencirati po njegovom originalnom nazivu ili aliasom. Npr. ako zadamo

[19] While, Dostupno na: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html

[20] Left Join, Dostupno na: https://www.sqltutorial.net/left-join.html

alias "b.name AS NazivAvioKompanije", unutar tog „querya“ možemo referencirati isti stupac kao 'b.name' ili 'NazivAvioKompanije'.



Showing rows 0 - 4 (5 total, Query took 0.0335 seconds.)

```
SELECT a.*, b.name as AvioKompanija FROM rezervacije as a LEFT JOIN AvioKompanije as b ON b.id = a.AvioKompanijeId
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

id	datum_polaska	vrijeme_polaska	hotel	broj_ljudi	ime_klijenta	kontakt_klijenta	AvioKompanijeId	vrijeme_dolaska	datum_dolaska	AvioKompanija
6	20.10.2019.	20:20	Sheraton	3	Maja Radic	0996523258	0			*NULL*
15	20.11.2019.	20:00	Lone Rovinj	3	Ana	Maric	9	21:00	25.11.2019.	Croatia
16	20.11.2019.	21:00	Hotel Lone	3	Ana	09985658	10	20:00	10.11.2019.	LH657
17	20.11.2019.	21:20	Lone, Rovinj	8	Maja Maric	099985621	9	20:10	10.11.2019.	Croatia
18	20.11.2019.	23:00	Lido, Rabac	3	Matija Novak	0985687458	11	21:20	11.11.2019.	Croatia LH568

Slika 9.- Prikaz upit „SELECT, LEFT JOIN“

Izvor: Autorski rad

Drugi dio "FROM rezervacije AS a LEFT JOIN AvioKompanije AS b" odnosi se na tablice iz kojih želimo dohvatiti stupce iz prvog djela „query-a“ - rezervacije AS a = odabir tablice rezervacije, te pridodajemo joj alias 'a'- AvioKompanije AS b = odabir tablice AvioKompanije, te pridodajemo joj alias 'b' - naredba (ako se može tako definirati) LEFT JOIN označava da se lijeva tablica (doslovno lijevo od same naredbe LEFT JOIN, tj. 'rezervacije' koristi kao 'glavna tablica' dok tablica AvioKompanije kao 'sporedna tablica'. To znači da će se ispisati svi redovi iz tablice 'rezervacije' (osim ako nije drugačije definirano sa „WHERE“, dok stupac 'AvioKompanijeId' iz 'rezervacije' koji služi kao referenca (tj. Foreign key) na drugu tablicu 'AvioKompanije', ispisat će samo podatke koji se poklapaju sa stupcem 'ID' iz tablice 'AvioKompanije' (što bi bilo "b.id = a.AvioKompanijeId", može se čitati isto kao "AvioKompanije.id=rezervacije.AvioKompanijeId"). Pošto je napisano 'b.name', prikazati će se naziv umjesto ID-a iz tablice 'AvioKompanije', koji se poklapa sa referenciranim ID-em (AvioKompanijeId) iz tablice 'rezervacije', ako nema poklapanja podatak je tipa NULL. (Npr. rezervacije.AvioKompanijeId je 99, a u tablici 'AvioKompanije' nema podatka sa ID-em 99, tad će rezultat biti NULL). Na slici broj 9. prikazano je kako se može provjeriti u bazi i konačni izgled nakon postavljana upita.

```
120             rezervacija.setVrijemeDolaska(rs.getString("vrijeme_dolaska"));
121             rezervacija.setHotel(rs.getString("hotel"));
122             rezervacija.setBrojLjudi(rs.getInt("broj_ljudi"));
123             rezervacija.setImeKlijenta(rs.getString("ime_klijenta"));
124             rezervacija.setKontaktKlijenta(rs.getString("kontakt_klijenta"));
125
126             AvioKompanija aviokompanija = new AvioKompanija();
127             aviokompanija.setName(rs.getString("AvioKompanija"));
128             rezervacija.setAvioKompanija(aviokompanija);
129
130             listaRezervacija.add(rezervacija);
131         }
132         // zatvaranje konekcije
133         conn.close();
134     } catch (ClassNotFoundException ex) {
135         Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
136     } catch (InstantiationException ex) {
137         Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
138     } catch (IllegalAccessException ex) {
139         Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
140     } catch (SQLException ex) {
141         Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
142     }
143
144     List<Rezervacija> filtriraneRezervacije = new ArrayList<>();
145
146     for (Rezervacija r : listaRezervacija) {
147         if (!klijent.isEmpty() && !klijent.equals(r.getImeKlijenta()))
148             continue;
149         if (!datumPolaska.isEmpty() && !datumPolaska.equals(r.getDatumPolaska()))
150             continue;
151         if (!vrijemePolaska.isEmpty() && !vrijemePolaska.equals(r.getVrijemePolaska()))
152             continue;
153         if (!datumDolaska.isEmpty() && !datumDolaska.equals(r.getDatumDolaska()))
154             continue;
155         if (!vrijemeDolaska.isEmpty() && !vrijemeDolaska.equals(r.getVrijemeDolaska()))
156             continue;
```

Slika 10.- Prikaz klase “DBHandler”- funkcija „dohvatRezervacijeIzBaze“ 2 dio

Izvor: Autorski rad

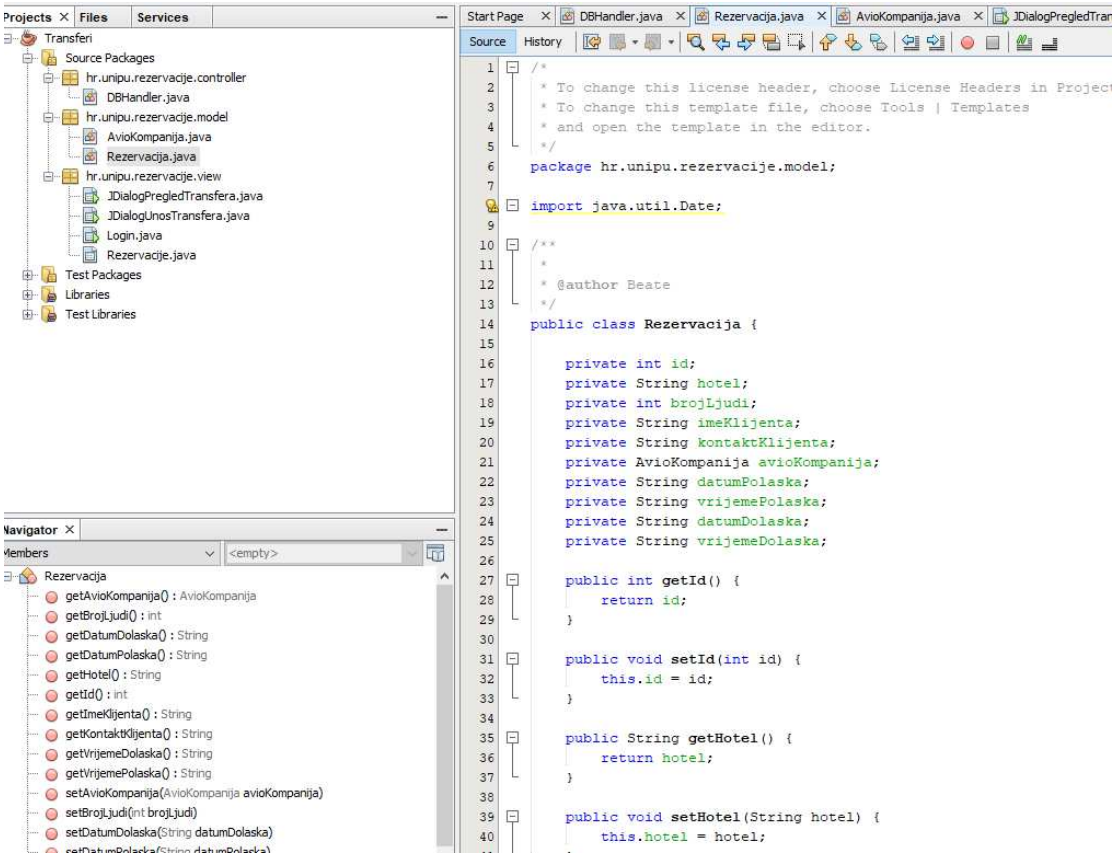
Na slici 10. imamo prikazanu „for“ petlju koja prolazi kroz listu svih rezervacija, i služi za unos u bazu. Uvjet „if“ (!klijent.isEmpty() && !klijent.equals (r.getImeKlijenta()))“ provjerava postoji li klijent s tim imenom ili da li je unešeni klijent različit od onog što već postoji. Ako klijent ne postoji, onda se u bazu zapiše pod novim brojem.

```
public static Rezervacija brisiIzBaze(int id)
{
    try{
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        // konekcijski objekt s parametrima za konekciju
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
        PreparedStatement stmt = conn.prepareStatement("delete FROM rezervacije where id = ?");
        stmt.setInt(1, id);
        // konekcija na bazu i slanje upita
        int rs = stmt.executeUpdate();


    } catch (ClassNotFoundException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(DBHandler.class.getName()).log(Level.SEVERE, null, ex);
    }
    return null;
```

Slika 11.- Prikaz klase “DBHandler”- funkcija „brisiIzBaze“

Izvor: Autorski rad

Zadnja funkcija prikazana na slici 11. u DBHandler prikazuje naredbe „delete“ koja služi za brisanje retka u tablici.



Slika 12.- Prikaz klase "Rezervacije.java"

Izvor: Autorski rad

Prikaz drugog paketa „hr.unipu.rezervacije.model" koja se sastoji od dvije klase nalazi se na slici 12. Prikazana je klasa pod nazivom „Rezervacija.java". U ovom paketu kreiramo model i postavljamo varijable modela („private int id" itd.). Potom kreiramo „set" i „get" konstruktore koji služe za pristupanje parametrima i vraćanje rezultata, te za upisivanje vrijednosti u polja. [21]

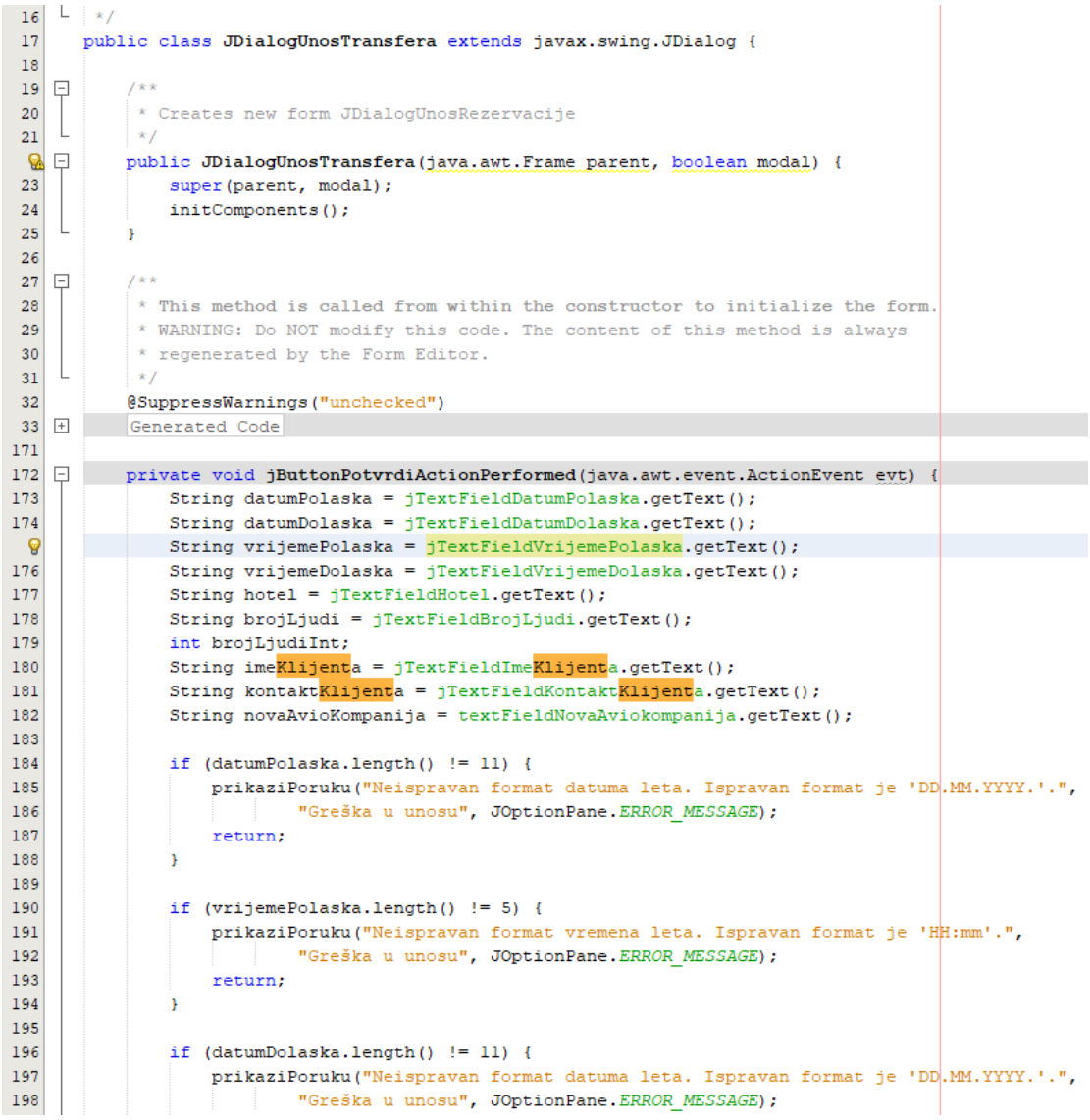
[21] Variables; Dostupno na:https://docs.oracle.com/javase/tutorial/java/javaOO/variables.html

```
 * @author Beate
 */
public class Login extends javax.swing.JFrame {

    private final static String USERNAME = "blazarev";
    private final static String PASSWORD = "pass";

    /**
     * Creates new form Login
     */
    public Login() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
       String username = jTextField1.getText();
       String password = jTextField2.getText();

       if (USERNAME.equals(username) && PASSWORD.equals(password)) {
           this.setVisible(false);
           new Rezervacije().setVisible(true);
       } else {
           jLabel2.setText("Unijeli ste krive korisničke podatke..");
       }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {...31 lines }
```

Slika 13.- Prikaz klase „Login.java“

Izvor: Autorski rad

Zadnji paket „hr.unipu.rezervacije.view“ se sastoji od četiri klase koje se odnose na izgled same aplikacije. Slika 13. prikazuje nam kodove u kojem smo definirali sva ograničenja za prijavu korisika u aplikaciju, tj. konačni izgled aplikacije. Dok povlačimo komponentu s palete na području dizajna, IDE automatski generira odgovarajući izvorni kod. Pod „public classom“ definirani su korisničko ime „username“ i lozinka „password“. Uvjet „if“ provjerava da li su podaci točno upisani. Ako je uvjet ispunjen, korisnik se prijavljuje u aplikaciju, a ako nije ispisati će se poruka da uvjet nije ispunjen.



```
 */
public class JDialogUnosTransfera extends javax.swing.JDialog {

    /**
     * Creates new form JDialogUnosRezervacije
     */
    public JDialogUnosTransfera(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButtonPotvrdiActionPerformed(java.awt.event.ActionEvent evt) {
        String datumPolaska = jTextFieldDatumPolaska.getText();
        String datumDolaska = jTextFieldDatumDolaska.getText();
        String vrijemePolaska = jTextFieldVrijemePolaska.getText();
        String vrijemeDolaska = jTextFieldVrijemeDolaska.getText();
        String hotel = jTextFieldHotel.getText();
        String brojLjudi = jTextFieldBrojLjudi.getText();
        int brojLjudiInt;
        String imeKlijenta = jTextFieldImeKlijenta.getText();
        String kontaktKlijenta = jTextFieldKontaktKlijenta.getText();
        String novaAvioKompanija = textFieldNovaAviokompanija.getText();

        if (datumPolaska.length() != 11) {
            prikaziPoruku("Neispravan format datuma leta. Ispravan format je 'DD.MM.YYYY.'.",
                    "Greška u unosu", JOptionPane.ERROR_MESSAGE);
            return;
        }

        if (vrijemePolaska.length() != 5) {
            prikaziPoruku("Neispravan format vremena leta. Ispravan format je 'HH:mm'.",
                    "Greška u unosu", JOptionPane.ERROR_MESSAGE);
            return;
        }

        if (datumDolaska.length() != 11) {
            prikaziPoruku("Neispravan format datuma leta. Ispravan format je 'DD.MM.YYYY.'.",
                    "Greška u unosu", JOptionPane.ERROR_MESSAGE);
```

Slika 14.- Prikaz klase „JDialogPregledTransfera.java“

Izvor: Autorski rad

Slika 14. nam prikazije još jednu klasu iz paketa pogleda, točnije klasu „Unos transfera“. Kao i na opisu prethodne slike 13. definirana su sva ograničenja za unos transfera u aplikaciju, tj. konačni izgled prozora za unos transfera. Također sadržava uvjet „if“, jedina razlika da ukoliko uvjet u ovom slučaju nije ispunjen, dolazi do iskočnog prozora sa porukom „Greška u unosu“. [22]

[22] Creatung gui; Dostupno na:

https://docs.oracle.com/javase/tutorial/uiswing/learn/creatinggui.html

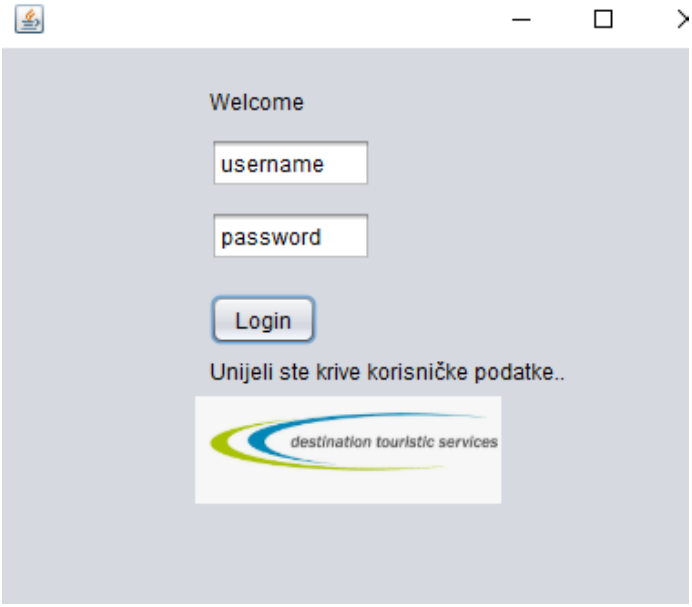
4. Primjena aplikacije

Za pokretanje aplikacije u alatnoj traci koristimo gumb „run“ zatim „run project“. Aplikacija se sastoji od četiri prozora, prvi „Login“ i drugi „Rezervacije“. Potom možemo odabrati želimo li unositi nove rezervacije za transfere „Unos Transfera“ ili pregledavati već postojeće rezervacije za transfere čemu služi četvrti prozor „Pregled Transfera“.

Slika 15. - Prikaz prozora „Log in“

Izvor: Autorski rad

Pokretanjem aplikacije otvara se prozor koji služi za prijavu korisnika „Login.java“ što je prikazano na slici 15. Da bi se korisnik uspješno ulogirao u sustav treba točno upisati korisničko ime i lozinku. Ako korisnik klikne na gumb „login“ i svi podatci su ispravno unešeni, korisniku se odobrava pristup aplikaciji. Ukoliko je unos neispravan, pojavljuje se poruka ispod „Login“ o pogrešnom unosu korisničkih podataka, prikazano na slici 16. „Unijeli ste krive korisničke podatke“ i pristup aplikaciji je odbijen.



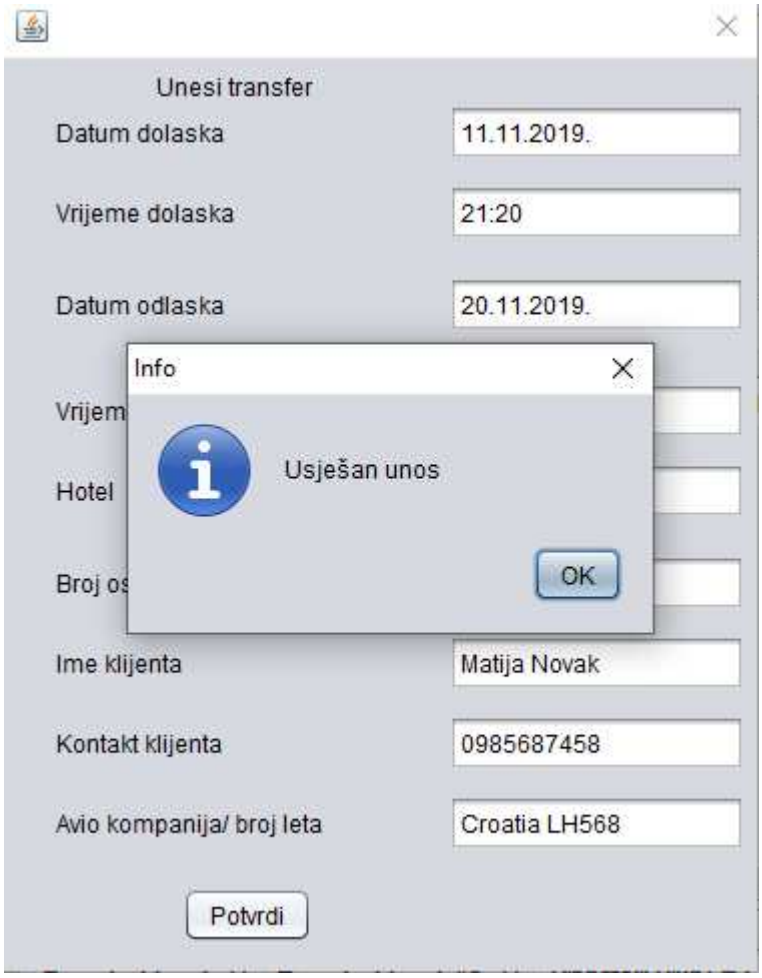
Slika 16. - Prikaz prozora „Log in“ sa porukom o pogrešnom unosu“

Izvor: Autorski rad

Slika 17. - Prikaz prozora „Rezervacije transfera“

Izvor: Autorski rad

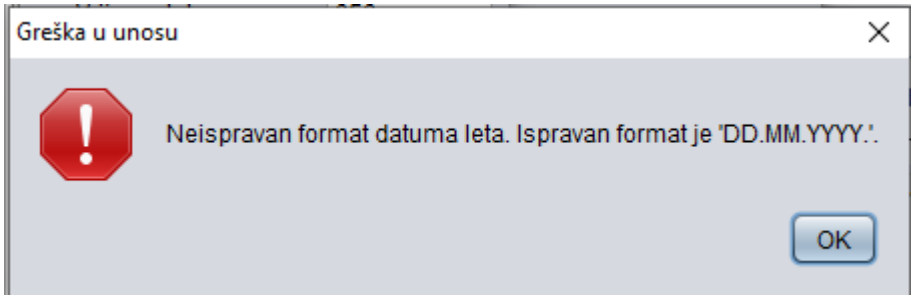
Nakon uspješne prijave u sustav, otvara se drugi ranije spomenut prozor. Korisnik može odabrati hoće li unijeti novi transfer u sustav ili će pregledati već postojeće klikom na gumb „unos transfera“ ili „pregled transfera“.



Slika 18. - Prikaz prozora „Unesi Trasfer“, s iskočnim prozorom „Uspješan unos“

Izvor: Autorski rad

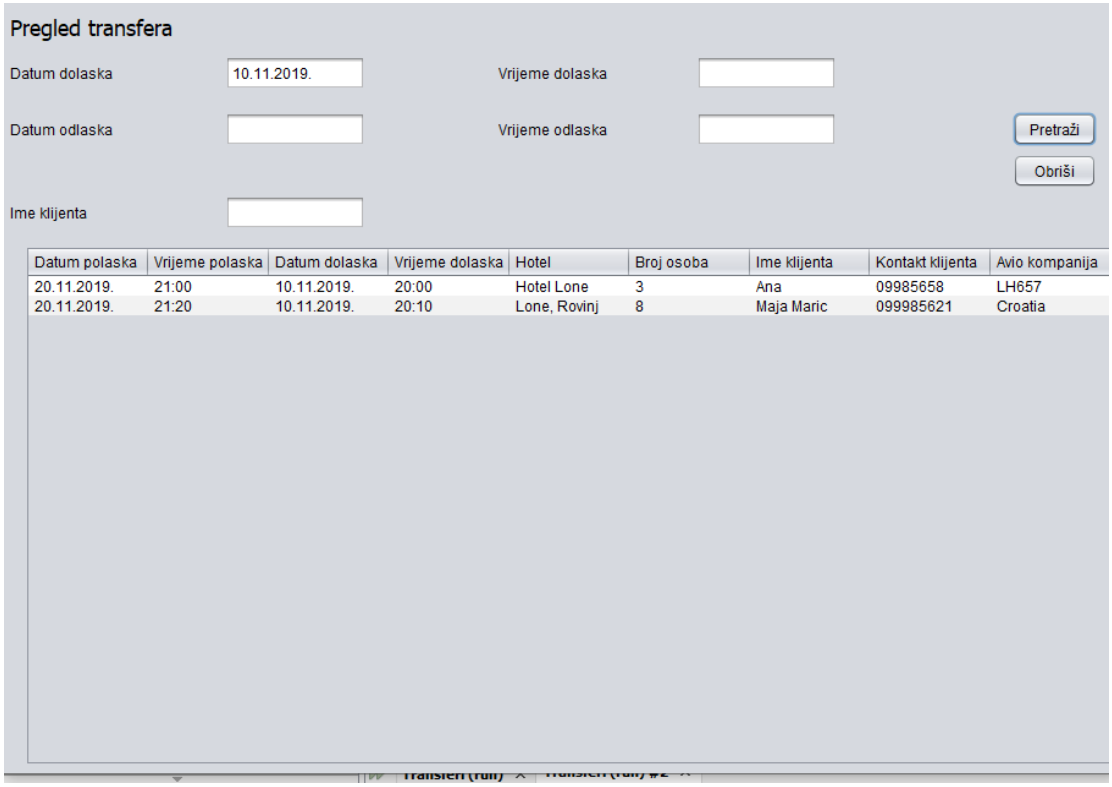
Pri unosu transfera korisnik mora točno ispisati podatke, kako bi unos bio uspješan. Tako za datum, vrijeme leta, broj osoba i kontakt klijenta može unijeti samo brojeve i to koji su točno određeni u Java klasi. Za „hotel“, „ime klijenta“ i „aviokompanija/broj leta“ može unositi bilo koje znakove. Nakon unosa klikom na gumb „Potvrdi“, aplikacija prikazuje novi prozor uz obavijest da je unos bio uspješan, kao što je prikazano na slici 18.



Slika 19. - Prikaz iskočnog prozora „Greška u unosu“

Izvor: Autorski rad

Ako je unos neispravan dolazi do greške pri unosu. Potom se pojavljuje novi iskočni prozor kao na slici 19., uz obavijest da je format neispravan i to točno za koji unos. Klikom na gumb „OK“ korisnik se vraća na prozor „Unesi Transfer“ te može ponovno unijeti ispravne podatke.



Slika 20. - Prikaz prozora „Pregled Transfera“

Izvor: Autorski rad

Korisnik kod pregleda može imati uvid u sve podatke u odabranom periodu tj. za određeni datum ili može pronaći određenog klijenta u sustavu ako su mu

potrebni podaci o konkretnom gostu. Podaci se spremaju u bazu, te prikažu u obliku tablice kao što je prikazano na slici 20. Korisnik može izmijeniti pogrešno unesene podatke klikom na stupac tj. polje koje želi izmijeniti, ili pak može izbrisati cijelu rezervaciju tako da je označi te klikom na gumb „Obriši“ obriše.

ZAKLJUČAK

Turizam kao brzorastući segment hrvatskog gospodarstva iz godine u godinu bilježi sve veću stopu rasta. Mnogi su upravo u ovom segmentu pronašli svoju priliku. Novi način poslovanja u dinamičnom okruženju, doveo je do potražnje za turističkim agencijama koje se bave turističkom destinacijom i čije je poslovanje specijalizirano za lokalna tržišta. Za uspjeh jedne takve turističke agencije važno je planirati, odrediti cilj i imati kvalitetne uvjete za rad. Budući da je potražnja sve veća, potrebno je nuditi sve širu paletu usluga, kako bi se privukli i najzahtjevniji gosti. Prateći ovaj trend, zaposlenici su često primorani u što kraćem vremenskom roku obraditi velike količine informacija.

Postupak izrade lista za transfere je prilično zamoran posao, prije svega zbog opširne papirologije i potrebe za izradom više tablica dnevno. Obzirom da je tehnologija prodrla u sve pore turističkog sektora, te da je koriste i kupci i zaposlenici tijekom i nakon putovanja, aplikacija koja pojednostavljuje i ubrzava prethodno pomenuti postupak mogla bi ponuditi korisno rješenje.

Sam naziv aplikacije "TRANSFERI" govori o čemu se točno radi. Ona nudi zanimljivo rješenje za zaposlenike u turističkoj destinaciji. Korisnik putem svog korisničkog računa može u aplikaciju unositi, pregledavati, izmijenjivati i brisati rezervacije. Nakon unosa nove rezervacije od strane korisnika, podaci se spremaju u bazu podataka, te se automatski spremaju i u tablicu. Ova aplikacija bi mogla biti od koristi i samim poslodavcima, budući da je koncipirana tako da u svakom trenutku ažurno pruža uvid u evidenciju dolazaka i odlazaka gostiju.

Prije izrade same aplikacije bilo je važno detaljno odrediti zahtjeve, potom definirati točne podatke koje će se prikupljati, te zatim prikazivati. Aplikacija je izrađena u Javi, programskom jeziku u razvojnom okruženju NetBeans. Odabran je zbog jednostavnosti i lake izrade forme, koja je neophodna za ovaj zadatak. Važan dio aplikacije je baza podataka, izrađena pomoću phpMyAdmin, u koju se spremaju svi generirani zahtjevi. Prilikom izrade desktop aplikacije, cilj je bio naučiti osnove programiranja, kreiranje MySQL

baze, te povezivanje aplikacije s bazom. Znanje stečeno tijekom izrade aplikacije vrlo je korisno i može se primjeniti pri rješavanju sličnih zadataka vezanih uz automatizaciju u poslovanju.

Kao i svaki programski proizvod, ova aplikacija nije savršena, ali ima potencijal za nadogradnju i ostavlja dovoljno prostora za proširenje. Moglo bi se poraditi na izradi vizualno modernijeg sučelja, spojiti s ostalim sustavima, te dodati opcije za „rent-a-car“ uslugu tijekom boravka u destinaciji. Otvorena je i mogućnost izrade mobilne aplikacije, koja bi bila dostupna i gostima tokom boravka, kako bi saznali npr. točne informacije o „pick-up“ terminu (točno vrijeme dolaska po gosta u hotel pri povratku). Budući da se radi o gostima koji tijekom boravka u destinaciji nisu mobilni, kao dodatna opcija moglo bi im se ponuditi i pružanje usluge transfera tijekom odmora. Ukoliko ova aplikacija zaživi, sigurno će biti od velike pomoći u informiranju o mnogim sadržajima koji su potrebni kako zaposlenicima tako i gostima. Putem osobnih računala i mobilnih uređaja ona podiže komunikaciju između zaposlenika na višu razinu, te bi se u budućnosti, naravno uz doradu, mogla iskoristiti i za povratne informacije na relaciji zaposlenik-gost i obratno.

SAŽETAK

Destinacijske turističke usluge su uz pomoć IT-stručnjaka uspostavile inteligentan, rekreativan „sve-u-jedan“ sustav, međutim još uvijek se trude poboljšati kvalitete uvjete rada zaposlenicima. Tema završnog rada je izrada aplikacije "TRANSFERI" koja služi za potporu upravljanja turističkom destinacijom. Aplikacija će uštedjeti na vremenu i nudi novi koncept poslovne automatizacije za usluge kojima raspolažu turoperatori i zaposlenici tvrtke. Razvijena je iz pažljivog promatranja i razumijevanja neprestanih potreba u turističkoj industriji. Definirani su zahtjevi korisnika i programski alati koji su korišteni za izradu iste. Desktop aplikacija razvijena je pomoću alata NetBeans, a kodovi su pisani JAVA jezikom. Baza je izrađena pomoću XAMPP web servera i softverskog besplatnog phpMyAdmin alata. Aplikacija se sastoji od četiri pogleda. Prvi se sastoji od „log-in“ prozora tj. služi za prijavu korisnika. Za odabir unos podataka ili pregled transfera je drugi prozor, dok treći služi za unos podataka, a četvrti za pregled transfera po datumu ili određene rezervacije.

Ključne riječi: phpMYadmin,XAMPP, desktop aplikacija, NetBeans, JAVA, destinacijski menadžment, specifikacija zahtjeva

POPIS LITERATURE

KNJIGE:

1) Dr. Sc. Sandra Čorak, Željko Trenzer (2014), Destinacijski menadžment kompanije [Pristupljeno 5. listopada 2019.]
2) Jović, A., Horvat, M. I Ivošević, D (2015.) Procesi programskog inženjerstva: Oblikovanje programske potpore, Zagreb [Pristupljeno 12. listopada 2019.]
3) Manger, R. i Mauher, M. (2011.) Programsko inženjerstvo. Zagreb:Algebra d.o.o [Pristupljeno 12. listopada 2019.]

INTERNET:

1) Destination touristic services, Dostupno na: https://dtservices.travel/destinations/ [Pristupljeno 5. listopada. 2019.]
2) DerTouristik, Dostupno na: http://www.dertouristik.com/de/ [Pristupljeno 5. listopada. 2019.]
3) O.p.i. online partner interface, Dostupno na: https://opi.dertouristik.com/ [Pristupljeno 5. listopada. 2019.]
4) PhpMyadmin, About; Dostupno na: https://www.phpmyadmin.net/ [Pristupljeno 7. listopada. 2019.]
5) NetBeans, Dostupno: https://netbeans.org/features/index.html [Pristupljeno 7. listopada. 2019.]
6) Jave SE, Dostupno: https://www.oracle.com/technetwork/java/javase/overview/index.html [Pristupljeno 7. listopada. 2019.]
7) Softversko inžinjerstvo; Dostupno na: http://web.studenti.math.pmf.unizg.hr/~manger/si/SI-vjezbe-03.pdf [Pristupljeno 11. listopada. 2019.]
8) XAMPP instalacija Apache web servera za MySQL, Dostupno na; https://www.hdonweb.com/programiranje/xampp-instalacija-apache-web-servera-za-mysql-i-php [Pristupljeno 1. studenog. 2019.]
9) SQL Tutorial, Dostupno na: https://www.sqltutorial.net/ [Pristupljeno 2. studenog. 2019.]

10) Java, Dostupno na: https://www.geeksforgeeks.org/java/ [Pristupljeno 7. listopada 2019.]
11) TheJava™Tutorials, Dostupno na: https://docs.oracle.com/javase/tutorial/uiswing/learn/index.html [Pristupljeno 3.studenog 2019.]

POPIS SLIKA

Slika 1. XAMPP Control Panel.....13

Slika 2.- Prikaz tablice iz baze „Rezervacije“.....14

Slika 3.- Prikaz tablice iz baze „AvioKompanije“.....14

Slika 4.- Prikaz projekta “Transferi”.....16

Slika 5.- Prikaz “Import” ……..……..……17

Slika 6.- Prikaz klase “DBHandler”- funkcija „unesiRezervacijuUBazu“.....18

Slika 7.- Prikaz klase “DBHandler”- funkcija „dohvatAvioKompanije“.....19

Slika 8.- Prikaz klase “DBHandler”- funkcija „dohvatRezervacijeIzBaze“ 20

Slika 9.- Prikaz upit „SELECT, LEFT JOIN“.....21

Slika 10.- Prikaz klase “DBHandler”- funkcija„dohvatRezervacijeIzBaze“ 2...22

Slika 11.- Prikaz klase “DBHandler”- funkcija „brisiIzBaze“.....23

Slika 12.- Prikaz klase “Rezervacije.java”.....24

Slika 13.- Prikaz klase „Login.java“.....25

Slika 14.- Prikaz klase „JDialogPregledTransfera.java“.....26

Slika 15. - Prikaz prozora „Log in“.....27

Slika 16. - Prikaz prozora „Log in“ sa porukom o pogrešnom unosu“.....28

Slika 17. - Prikaz prozora „Rezervacije transfera“.....28

Slika 18. - Prikaz prozora „Unesi Trasfer“, s iskočnim prozorom „Uspješan unos“.....29

Slika 19. - Prikaz iskočnog prozora „Greška u unosu“.....30

Slika 20. - Prikaz prozora „Pregled Transfera“.....30

SUMMARY

Destination touristic services have, with the help of IT experts, established an intelligent, recreational all-in-one system, but are still striving to improve the quality of working conditions for employees. The theme of the final thesis is the development of the application "TRANSFERI" which is used to support the management of the tourist destination. The application will save time and will offer a new business automated concept for the services of tour operators and company employees. Developed from careful observation and understanding of the continuing needs in the tourism industry. User requirements and software tools that were used to create the same are defined. The desktop application was developed using the NetBeans tool and the codes are written in the JAVA language. The database was created using XAMPP web server and the free software PhpMyAdmin tool. The application consists of four views. The first one is from the log-in window, which means it is used to log in the user. Second window is used for selecting data entry or transfer view, while the third one is specifically for data entry and the fourth window is used for the preview of the transfers or for a specific reservation.

Keywords: phpMyAdmin, XAMPP, desktop application, NetBeans, Java script, destination management, request specification