APLIKACIJA ZA POMOĆ U RADU UPRAVLJANJEM LJUDSKIM POTENCIJALIMA

Cmrk, Marko

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:473548

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-19**

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

APLIKACIJA ZA POMOĆ U RADU UPRAVLJANJEM LJUDSKIM POTENCIJALIMA

Marko Cmrk

Zagreb, veljača 2019.

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, 26.2.2019.

Predgovor

Ponukan nemalim vlastitim iskustvom rada u više različitih kompanija s različitim stupnjem sofisticiranosti sustava za evidenciju dolazaka i odlazaka zaposlenika i posljedica koje to donosi, te komunikacijom s poznanicima koji su se susretali s istim situacijama, dobio sam ideju za više-manje generičko rješenje te problematike. S vremenom ta je ideja prerasla u cjelokupno rješenje za pomoć pri radu s ljudskim potencijalima. Primjer tog rješenja odlučio sam predložiti kao temu ovog završnog rada. Ovim putem želio bih se zahvaliti svim kolegama, poznanicima i mentoru koji su mi svojim uvidima i savjetima pomogli razraditi detalje rješenja te tako olakšali razvoj aplikacije i pisanje rada.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Ideja ovog završnog rada je obraditi problematiku vezanu za upravljanje ljudskim potencijalima u tvrtkama i kompanijama. Većina poslovnih procesa u odjelima za upravljanje ljudskim potencijalima još se uvijek odrađuje ručno i time bespotrebno usporava obradu pristiglih zahtjeva. Navedeni su procesi veoma pogodni za neki oblik automatizacije koji bi ih modernizirao, ubrzao i povezao s ostalim sustavima unutar tvrtke, što se ovim radom i pripadajućom aplikacijom želi demonstrirati.

Ključne riječi: ljudski potencijali, poslovni proces, ručno, automatizacija.

Abstract

The idea of this final paper is to address the issue of human resources management in small and large companies. Most of the business processes in human resource management departments are still handled manually and thus unnecessarily slow down the processing of received requests. Those processes are very suitable for some form of automation that would modernize, accelerate and connect them with other systems within the company, demonstrating which is the goal of this paper and adjacent application.

Keywords: human resources, business process, manual, automation.

Sadržaj

1. Uvod 1

2. Osnovne karakteristike aplikacije 2

2.1. Korištene tehnologije i prakse 2

2.2. Arhitektura sustava 4

2.3. Desktop aplikacija 5

2.3.1. Tehnologije desktop aplikacije 6

2.4. Web portal 6

2.4.1. Tehnologije web portala 7

2.5. Simulator evidencije radnih sati 10

3. Rad aplikacije 12

3.1. Rad unutar desktop aplikacije 12

3.1.1. Zaposlenici 13

3.1.2. Odsustva 16

3.1.3. Statistike 17

3.2. Rad u web portalu 20

3.3. Prednosti rada u aplikaciji 24

Zaključak 26

Popis kratica 27

Popis slika 28

Popis kôdova 29

Literatura 30

1. Uvod

Upravljanje ljudskim potencijalima ili kako to u mnogim kompanijama nazivaju „Human resources“ (u daljnjem tekstu **HR**), danas je jedan od najvažnijih sektora svake tvrtke ili kompanije koja po broju zaposlenih nadilazi opseg obiteljske proizvodnje. Globalni smjer poslovanja prepoznao je zaposlenika kao najvredniji resurs kompanije te da je dugoročno puno isplativije razvijati i ulagati u zaposlenika, nego ga mijenjati čim se ukaže potreba. To ulaganje podrazumijeva dodatno educiranje, brigu o potrebama zaposlenika, pribavljanje povratnih informacija od zaposlenika itd. Takvim pristupom osigurava se bolja usredotočenost zaposlenika, a samim time i kvalitetniji rad, tj. proizvod ili uslugu za koje je zaposlenik zadužen. Glavnu ulogu u navedenoj interakciji prema zaposlenicima ima HR. Iz tog razloga iznimno je bitno da taj odjel ima primjerenu informatičku podlogu i podršku kako bi svi podaci o zaposlenicima bili aktualni, točni i u skladu s podacima iz drugih odjela.

Kako prilikom kreiranja informatičkog sustava kompanije prednost imaju dijelovi i odjeli koji su jezgra poslovanja, često se dogodi da „pomoćni“ sustavi ne dobiju angažman koji trebaju i zaslužuju. To nerijetko rezultira prilično nepotpunim, dotrajalim rješenjima koja konstantno čekaju na nadogradnje koje nikako da dođu na red. Posljedica toga su aplikacije koje nisu stabilne, ni adekvatno povezane s ostatkom sustava pa ostavljaju puno posla za ručnu obradu. HR aplikacije se često susreću s tim problemom, bez obzira na poslovnu važnost samog odjela. Kvalitetno aplikativno rješenje za HR odjel indirektno može utjecati na zadovoljstvo samih zaposlenika te tako pospješiti cjelokupno poslovanje kompanije.

Kao posljedica svih navedenih razloga, ovim radom želi se predstaviti primjer generičkog rješenja HR aplikacije koje se lako može integrirati u postojeće poslovne sustave, pruža velik broj mogućnosti koje su potrebne u svakom HR-u, a jednako lako se može nadograditi dodatnim modulima.

U poglavlju „Osnovne karakteristike aplikacije“ generalno je opisana aplikacija, njena arhitektura, kao i osnovne informacije o korištenim tehnologijama i praksama, dok su u poglavlju „Rad aplikacije“ detaljno opisane funkcionalnosti i mogućnosti svih dijelova aplikacije.

2. Osnovne karakteristike aplikacije

Ono što ovu aplikaciju čini dobrim HR rješenjem jest činjenica da se njome dobiva centar informacija o svakom zaposleniku od kojeg svi ostali odjeli, npr. računovodstvo, mogu primati podatke.

Glavna karakteristika ove aplikacije je generičnost. Ona proizlazi iz činjenice da aplikacija sadrži sve osnovne funkcionalnosti koje pokrivaju glavne značajke radnog odnosa što podrazumijeva informacije o zaposleniku, evidenciju radnih sati, evidenciju odsustava i slično. Integracija ovog rješenja u postojeći sustav svedena je na najjednostavniju razinu koristeći sve tehnološke standarde umrežavanja i dijeljenja podataka. Istovremeno ta generičnost omogućuje iznimno laku nadogradnju i/ili dodavanje modula prema specifičnim zahtjevima pojedinog klijenta.

2.1. Korištene tehnologije i prakse

Cijela aplikacija bazira se na verziji **8 Java** programskog jezika.

Java programski jezik razvijen je 1995. godine od strane Jamesa Arthura Goslinga koji je u to vrijeme radio za tvrtku Sun Microsystems. Radi se o objektno orijentiranom jeziku koji je danas jedan od najpopularnijih i najraširenijih programskih jezika na svijetu. Koristi se u velikom rasponu uređaja i sistema, od igraćih konzola, osobnih računala do podatkovnih centara i superračunala. Od 2009. godine Java je u vlasništvu tvrtke Oracle Corporation. Trenutno aktualna verzija Jave je 11 iako je verzija 8 i dalje široko podržana i korištena. Postoji više izdanja Jave, ovisno o okolini za koju se razvija, a to su:

- Java Card – za pametne kartice
- Java ME (Micro Edition) – za okoline limitiranih resursa
- Java SE (Standard Edition) – za standardne razvojne okoline
- Java EE (Enterprise Edition) – za poduzeća i internet

Aplikacija koristi EE izdanje zbog dodatnih mogućnosti koje to izdanje sadrži, a koje su potrebne za kvalitetan rad aplikacije i njezinu integraciju na različite platforme i postojeće sisteme. (Wikipedia, Java (programming language), 2019; Oracle Corporation, 2019)

U pozadini aplikacije nalazi se **Microsoft SQL Server 2014** baza podataka.
Microsoft SQL Server je relacijska baza podataka koju je tvrtka Microsoft plasirala na tržište 1989. godine. Koristi jezik transakcijskih strukturiranih upita (engl. *Transactional Structured Query Language*, skraćeno T-SQL) što znači da osim osnovnih SQL upita dozvoljava i programske naredbe. Korištenje tog jezika i sama činjenica da se radi o relacijskoj bazi podataka podrazumijeva da se podaci u ovoj bazi spremaju u tablice s imenovanim kolonama. Relacijske baze podataka danas čine većinu baza u korištenju, a Microsoft SQL Server baza podataka jedna je od vodećih u svijetu. Tome svakako pridonosi činjenica da Microsoft ugrađuje podršku za spajanje na SQL Server baze podataka u sve svoje razvojne alate i okruženja, a pomoću ODBC *drivera* koje redovito isporučuje omogućuje i svim ostalim alatima lako i nesmetano spajanje. Trenutno aktualna verzija je SQL Server 2017, no Microsoft aktivno podržava sve verzije od SQL Server 2008. (TechTarget, Microsoft SQL Server, 2019)

Iako rješenje dolazi s navedenom bazom, jednostavnom migracijom podataka moguće je aplikaciju prilagoditi za rad s već postojećom bazom podataka što je još jedan dokaz već spomenute generičnosti aplikacije. Ta generičnost proizlazi iz korištenja dobrih praksi programiranja koje su prisutne kroz cijelu aplikaciju. Specifično, za povezivanje svih dijelova aplikacije s bazom podataka korišten je ***singleton pattern***.

Singleton pattern je princip kodiranja koji dozvoljava samo jednu instancu pojedine klase. Dolazi od matematičkog pojma gdje se skup sastoji od jednog elementa, a koristi se najčešće u situacijama gdje jedan objekt treba biti zadužen za koordinaciju radnji cijelog sustava. (Wikipedia, Singleton pattern, 2019)

```
public class DataSourceSingleton {
    private static final String SERVER_NAME = "****";
    private static final String DATABASE_NAME = "****";
    private static final String USER = "****";
    private static final String PASSWORD = "****";

    private DataSourceSingleton() {}

    private static DataSource instance;
    public static DataSource getInstance() {
        if (instance == null) {
            instance = createInstance();
        }
```

```
            return instance;
        }
        private static DataSource createInstance() {
            SQLServerDataSource dataSource = new
SQLServerDataSource();
            dataSource.setServerName(SERVER_NAME);
            dataSource.setDatabaseName(DATABASE_NAME);
            dataSource.setUser(USER);
            dataSource.setPassword(PASSWORD);
            return dataSource;
        }
}
```

Kôd 2.1. *Singleton pattern* za pristup bazi podataka iz aplikacije

U ovoj situaciji *singleton pattern* brine da se npr. u slučaju gdje HR i zaposlenik istovremeno ažuriraju podatke te ih pokušavaju pohraniti, podaci pohrane na pravilan način tako da dozvoljava komunikaciju prema bazi samo preko jedne instance klase. U kôdu to izgleda kao što je prikazano kroz Kôd 2.1.

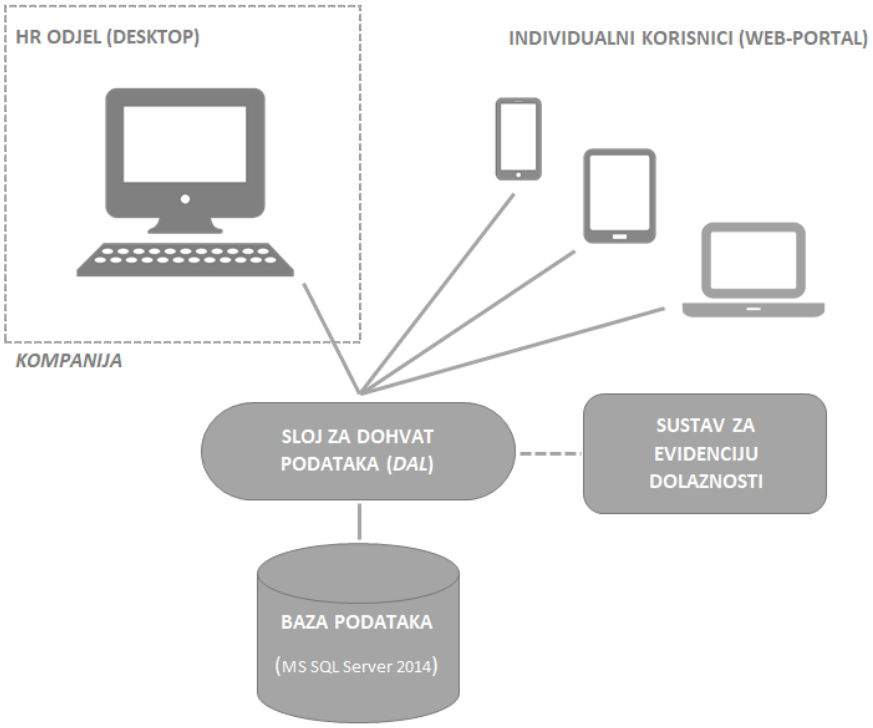
Još jedna praksa dobrog programiranja korištena kroz aplikaciju je odvajanje funkcijskih slojeva (pristup bazi podataka, poslovna logika, grafičko sučelje). To je dodatno naglašeno posebno izdvajanjem cijelog sloja za pristup bazi podataka kojeg pomoću ugrađenih sučelja koriste svi ostali dijelovi aplikacije.

Single responsibility principle je princip programiranja koji diktira da svaki dio funkcionalnosti softvera ima jednu dediciranu klasu za taj dio te je ta klasa u potpunosti odgovorna za pravilan rad tog dijela. (Toptal, Single responsibility principle, 2019) Sve klase unutar ovog sustava prate taj princip i tako omogućuju iznimnu modularnost. Kao što se vidi kroz Kôd 2.1., klasa za pristup bazi podataka uz to što implementira *singleton pattern*, također implementira i *single responsibility pattern*.

2.2. Arhitektura sustava

Aplikacija je podijeljena na dva osnovna dijela; stolnu (engl. *desktop*) aplikaciju i web portal. Desktop aplikacija je okosnica ovog rješenja u smislu da se kroz nju odrađuje većina zadaća HR odjela. Smješta se lokalno, unutar tvrtke te time dozvoljava pristup samo ovlaštenim djelatnicima. Web portal je personalizirani portal koji služi individualnim zaposlenicima za prikaz i ažuriranje osobnih informacija i statusa. Uz osnovnu aplikaciju

dolazi i pomoćni simulator evidencije radnih sati. On služi za simulaciju navedene evidencije u slučaju kad aplikacija nije integrirana u realni sustav. Također, može poslužiti kao predložak ili idejno rješenje ukoliko kod klijenta nije predviđen sustav za evidenciju dolaznosti. Slika 2.1. grafički prikazuje shemu arhitekture sustava i međusobnog odnosa pojedinih dijelova aplikacije.



Slika 2.1. *Shema* arhitekture sustava

2.3. Desktop aplikacija

Kao što je već napomenuto, desktop aplikacija je okosnica i temelj ovog rješenja. Služi kao centralni dio pomoću kojeg nadležni zaposlenici HR odjela imaju pristup svim relevantnim podacima o zaposlenicima kompanije te ih po potrebi mogu ažurirati.

Osnovne funkcionalnosti ovog dijela su uvid i manipulacija osnovnim podacima zaposlenika, evidencija radnih sati, evidencija odsustva te uvid u hijerarhijski poredak kompanije. To su elementi koji su esencijalni za HR odjel svake kompanije, dok je ostavljena mogućnost dodavanja dodatnih modula kako bi se proširila funkcionalnost aplikacije prema specifičnim zahtjevima klijenta.

Činjenica da je aplikacija razvijena pomoću Java programskog jezika ne uvjetuje nikakvo posebno operativno okruženje, tj. radi na svakoj platformi koja je u mogućnosti pokretati

Java aplikacije. Kako se aplikacija postavlja lokalno na računala zaposlenika HR-a, nema potrebe za dodatnim sistemom autentifikacije unutar same aplikacije, nego se za provjeru identiteta koristi autentifikacija samog operacijskog sustava, npr. *Active Directory* u Windows okruženju. Aplikacija koristi taj identitet da bi pratila tko koristi aplikaciju te, još bitnije, tko unosi promjene. Svaka promjena podataka spremljena je u bazu u obliku *logova* kako bi se otklonila mogućnost malverzacije podacima.

2.3.1. Tehnologije desktop aplikacije

Dodatna tehnologija u desktop aplikaciji koja se koristi uz Java programski jezik jest **JavaFX**.

Radi se ekstenziji same Jave koja omogućuje razvoj desktop aplikacija, točnije njihovih grafičkih sučelja. Izdana je 2008. godine kao zaseban paket, no od Java verzije 8 postala je integralni dio jezika. Najvažniji dio JavaFX tehnologije je **Scene Builder**, grafičko sučelje platforme pomoću kojeg je moguće izraditi grafička sučelja aplikacija koje se razvijaju. Taj alat je iznimno bitan jer zbog činjenice da u realnom vremenu prikazuje kako će sučelje izgledati i mogućnosti da se njime manipulira, uvelike olakšava i ubrzava razvoj. Mnoge Java razvojne okoline (engl. *Integrated Development Environments*, skraćeno IDE) danas u sebi imaju integriran neki oblik Scene Bodabranom filteru u polju uildera te tako pružaju iznimno intuitivan i lagodan proces razvoja aplikacija. (Oracle, Java Documentation, 2019)

Svi vidljivi elementi aplikacije mogući su zahvaljujući JavaFX ekstenziji koja radi na principu povezivanja tih elemenata, tj. radnji za koje se očekuje da pojedini element odradi, s kôdom u pozadini aplikacije.

2.4. Web portal

Što se tiče svih zaposlenika, osim onih koji imaju pristup desktop aplikaciji, web portal je jedini kontakt zaposlenika sa cijelim HR sustavom. Portal je individualno personaliziran za svakog zaposlenika te mu pruža mogućnosti nadzora i kontrole nad vlastitim podacima vezanim za HR, kao što su evidencija dolazaka i odlazaka, prikaz prekovremenih sati, mogućnost upravljanja vlastitim odsustvima i osobnim podacima te, ukoliko je zaposlenik voditelj nekog od timova, ažuriranje pristiglih zahtjeva za odsustvima. Postavljanje portala kao web aplikacije omogućuje pristup s velikog broja uređaja te ne ograničava pristup na

vrijeme ni mjesto. Drugim riječima, svaki aktivni zaposlenik kompanije može pristupiti portalu svugdje i bilo kad putem web preglednika na svom uređaju. Navedena mogućnost je iznimno bitna u situacijama kao što su bolovanja ili neki drugi neplanirani izostanci s posla. U takvom slučaju voditelj zaposlenika, kao i HR, odmah su upoznati sa stanjem te tako mogu pravodobno i adekvatno reagirati ukoliko za to postoji potreba. Portal se također koristi kako bi voditelji timova odobravali ili odbijali zahtjeve za odsustvima njihovih podređenih. Nakon što zaposlenik napravi zahtjev za odsustvom taj zahtjev je istog trena vidljiv njegovoj nadređenoj osobi, kao i HR osoblju kroz desktop aplikaciju. Isto tako zaposlenik vidi statuse svojih zahtjeva.

U dogovoru s ostalim nadležnim odjelima i vodstvom, moguće je nadograditi web portal da prikazuje više od podataka usmjerenih na HR. Primjer toga bi bio dodatni *tab* s prikazom platnih lista i/ili nagrada koje je zaposlenik u prošlosti primio. Na taj način bi web portal mogao postati centralno mjesto preko kojeg zaposlenik ima uvid u sve svoje podatke vezane za kompaniju.

2.4.1. Tehnologije web portala

Uz spomenute tehnologije koje su korištene kroz cijelu aplikaciju u razvoju web portala vrijedi spomenuti i sljedeće tehnologije.

Spring Framework je iznimno popularan razvojni okvir (engl. *framework*) za Java aplikacije. Na tržištu se pojavio 2002. godine i do danas broji pet osnovnih verzija. Radi se o *frameworku* najčešće korištenom u razvoju web aplikacija zbog jedne od svojih osnovnih karakteristika, a to je *dependency injection*. To znači da se *framework* brine o konfiguraciji i upravljanju Java objektima pomoću refleksije. Takav pristup uvelike olakšava razvoj jer miče brigu o životnim ciklusima pojedinih objekata s programera i ostavlja više prostora i vremena za bavljenje poslovnom logikom i funkcionalnostima same aplikacije. (Spring Framework, 2019; Spring, 2019)

Najviše korištena grana Spring *frameworka* u ovoj aplikaciji je Spring Web MVC koji omogućuje rukovanje osnovnim klasama (modelima) iz Java kôda te njihov prikaz i manipulaciju na web stranicama pomoću upravljača (engl. *controller*). Ovim putem Spring *framework* još jednom olakšava razvoj preuzimajući na sebe kontrolu izvršavanja i rukovanja klasama kako programer ne bi morao paziti na svaki korak. (Spring, Web on Servlet Stack, 2019)

JavaScript je skriptni programski jezik koji se izvršava u web pregledniku na strani korisnika. Razvila ga je tvrtka Netscape 1995. godine, a trenutno je u vlasništvu Oracle Corporation-a, kao i Java programski jezik. Iako je rađen po uzoru na Javu, ne radi se o čisto objektno orijentiranom jeziku, nego umjesto klasa koristi prototipove. Trenutno aktualna verzija je 1.8.5 koja je podržana u velikom broju najčešće korištenih web preglednika. (MDN web docs, 2019)

```
function openTab(evt, tabName) {
    // Declare all variables
    var i, tabcontent, tablinks;

    // Get all elements with class="tabcontent" and hide them
    tabcontent =
document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
}
```

Kôd 2.2. Primjer funkcije pisane u JavaScript programskom jeziku

Navedena funkcija (Kôd 2.2.), ovisno o selektiranom elementu (gumbu), prikazuje sadržaj pripadajućeg *taba*, dok sakriva sadržaje ostalih *tabova*.

Bootstrap je još jedan iznimno popularan i često korišten *framework*, no ovog puta radi se o *front-end frameworku.* Pojavio se 2011. godine i od tada je nezaobilazan dio razvoja svake web aplikacije zbog svoje sveobuhvatnosti i podrške za veliku većinu *core* web tehnologija kao što su HTML i CSS. U osnovi se sastoji od velike količine stilova primjenom kojih bazična web stranica poprima puno profesionalniji i oku ugodniji izgled. Uz osnovni paket, na internetu postoji i velik broj dodatnih stilova kreiranih od strane korisnika. Upravo u tome leži snaga ovog *frameworka*. Kako se radi o softveru otvorenog kôda, svi su korisnici u mogućnosti plasirati vlastite stilove na veliku količinu dediciranih repozitorija na internetu te se tako dolazi do skoro beskonačnog broja kombinacija stilova među kojima svatko može pronaći rješenje za svoje potrebe. Jedna od bitnih stavki ovog *frameworka* je i automatska podrška za responzivnost web stranica koja je u današnje vrijeme kad se internet koristi na ogromnom broju različitih uređaja, iznimno vrijedna karakteristika uvelike ubrzavajući i olakšavajući razvoj. Uz vizualne, Bootstrap donosi i mnoge funkcionalne napretke koristeći unaprijed isprogramirane JavaScript funkcije. Čak i u slučaju specifičnih situacija za koje je prilično teško naći apsolutno adekvatno rješenje na

internetu, lako je prilagoditi neke od postojećih funkcija. (Wikipedia, Bootstrap (front-end framework), 2019; Introduction – Bootstrap, 2019)

Korištenjem Bootstrap *frameworka* uvelike se smanjuje vrijeme potrebno za razvoj web aplikacija zbog principa „ako već postoji rješenje problema, nije ga potrebno ponovo tražiti“.

Apache Tomcat je HTTP web poslužitelj (engl. *server*) koji omogućuje posluživanje web aplikacija pisanih Java programskim jezikom. Plasiran je na tržište 1999. godine od strane tvrtke Sun Microsystems. Trenutno aktualna verzija je 9.0, iako se za ovu aplikaciju koristi verzija 8.5 koja podjednako podržava Java verziju 8. (TheServerSide, 2019)

Tomcat je softver otvorenog kôda čiji je razvoj produkt kolaboracije velikog broja vrhunskih programera. Iznimno je raširen i koristi se u širokoj paleti različitih upotreba. (Apache Tomcat, 2019)

Unutar ove aplikacije Tomcat se koristi u svrhu posluživanja web portala. Odabran je zbog toga što predstavlja tehnološki standard u svijetu te pruža podršku za veliki broj različitih okolina. Dakako, prilikom implementacije rješenja kod klijenta potrebno je utvrditi je li Tomcat najbolji izbor za integraciju unutar njihove postojeće okoline te provjeriti alternative. Alternativa može biti bilo koji *server* s mogućnošću posluživanja Java web aplikacija koje sadrže JSP datoteke.

JSP, skraćenica od JavaServer Pages, je ekstenzija Java programskog jezika koja omogućuje razvoj dinamički generiranih web stranica baziranih na HTML, XML ili sličnim jezicima koristeći dijelove kôda pisanog u Javi. Tvrtka Sun Microsystems JSP tehnologiju objavila je na tržištu 1999. godine i do danas se razvilo nekoliko verzija od kojih je trenutno aktualna 2.3. JSP nudi mogućnost izvođenja dijelova Java kôda (*scriptleta*) na samoj web stranici. *Scriptleti* se u kôdu označavaju pomoću posebnog znakovlja kao što je vidljivo na primjeru (Kôd 2.3.). Kako se izvođenje kompleksnijih dijelova kôda kosi sa spomenutim principima dobrog kodiranja, korištenje *scriptleta* u JSP datotekama u praksi je svedeno na minimum, tj. na korištenje osnovnih programskih funkcija kao što su *if* blokovi ili petlje. (Wikipedia, JavaServer Pages, 2019; Kučak, 2013)

```
<p>Counting to three:</p>
<% for (int i=1; i<4; i++) { %>
    <p>This number is <%= i %>.</p>
<% } %>
<p>OK.</p>
```

Kôd 2.3. Primjer petlje korištenjem *scriptleta* u JSP datoteci

Da bi se izbjeglo pretjerano korištenje *scriptleta* u JSP datotekama, a zadržala i poboljšala funkcionalnost tehnologije, razvijene su službene biblioteke koje implementiraju veliku količinu različitih, često korištenih funkcija. Te biblioteke poznate su pod nazivom JavaServer Pages Standard Tag Library, ili skraćeno **JSTL**. Da bi se biblioteke koristile potrebno ih je uključiti u JSP datoteku na sljedeći način:

```
<%@taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core"%>
```

Postoji pet osnovnih biblioteka od kojih svaka pokriva funkcije određene domene. Najznačajnija je *Core* biblioteka koja sadrži osnovne Java programske funkcije. Slično *scriptletima,* i ove biblioteke označavaju se posebnim znakovljem kao što je prikazano u primjeru (Kôd 2.4.). (TutorialsPoint, 2019; Kučak, 2013)

```
<c:if test = "${brStudenata >= 50}">
         <p> <c:out value = "Vrlo dobra popunjenost"/><p>
</c:if>
```

Kôd 2.4. Primjer *if* bloka korištenjem JSTL *Core* biblioteke

2.5. Simulator evidencije radnih sati

Simulator evidencije radnih sati je pomoćni modul koji prvenstveno služi za prezentacijske svrhe sustava. Ukoliko kompanija koja implementira ovo rješenje već ima svoj sustav evidencije radnih sati, tj. dolazaka i odlazaka zaposlenika, aplikacija se tom sustavu jednostavno može prilagoditi. U protivnom simulator može poslužiti kao idejno rješenje za takav sustav koji je vitalan u svakoj kompaniji za praćenje produktivnosti i zalaganja. Na taj način dobivaju se jednoznačni podaci o svakom zaposleniku. U realnoj situaciji može ga zamijeniti npr. sistem evidencije provlačenjem personalizirane kartice kroz uređaj na ulazu u zgradu ili odjel. Naprednije verzije mogu biti bazirane na biometrijskim oznakama

ili drugim oblicima prepoznavanja, no osnovna funkcionalnost ostaje ista, a to je bilježenje vrijeme dolazaka i odlazaka pojedinačnog zaposlenika.

Simulator radi na principu upisivanja broja radnika za kojeg se želi odraditi evidenciju dolaska/odlaska. Pritiskom na označeni gumb (Slika 2.2.) odrađuje se upis evidencije u bazu podataka. Upis sadrži sve potrebne podatke: tko, kada i radi li se o dolasku ili odlasku. Ukoliko u bazi postoji evidencija dolaska određenog zaposlenika, a koja nije uparena odlaskom, moguće je evidentirati samo odlazak. Takva rigorozna logika za praćenje navedene aktivnosti sasvim je dovoljna jer bilo kakva odstupanja od osnovnog pravila mogu biti posljedica jedino iznimnih situacija koje je u tom slučaju svakako potrebno iskomunicirati HR odjelu. Kao idejno rješenje, ovo je podložno daljnjoj nadogradnji i mogućoj nadopuni prema zahtjevima i specifičnim potrebama klijenta.

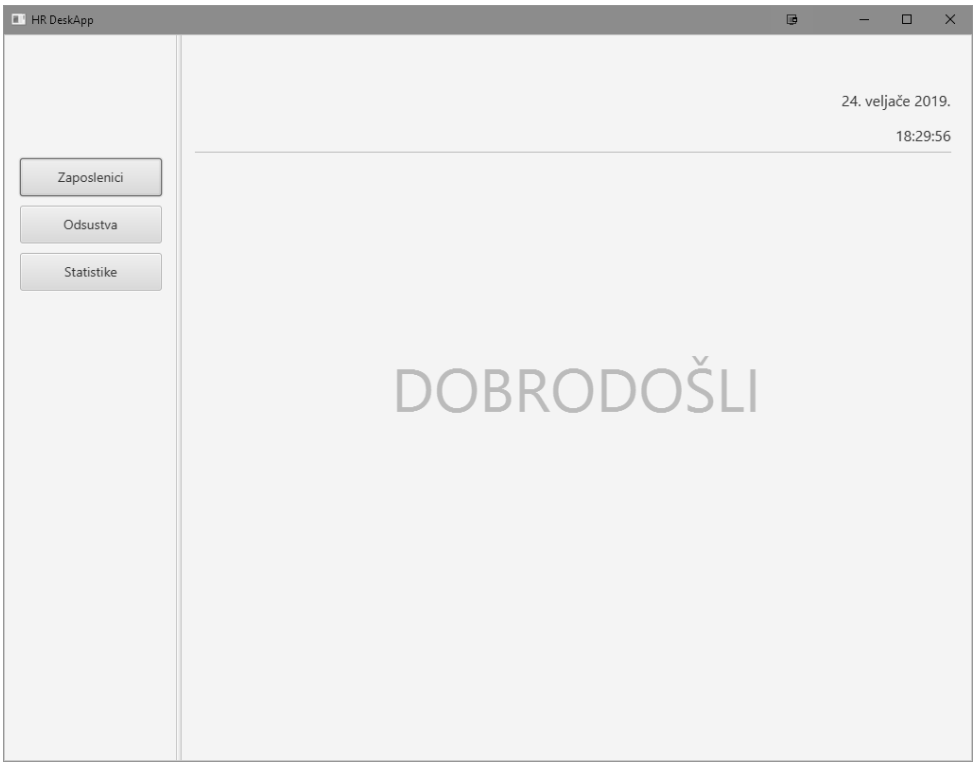
Kao i kod desktop aplikacije, za razvoj grafičkog dijela simulatora korištena je JavaFX tehnologija, dok je za sam kôd korištena verzija 8 programskog jezika Java.

Slika 2.2. Simulator evidencije dolazaka i odlazaka zaposlenika

3. Rad aplikacije

3.1. Rad unutar desktop aplikacije

Desktop aplikacija podijeljena je u tri okvira koji korisnika aplikacije intuitivno navode kroz ekrane prema radnjama koje korisnik želi obaviti.

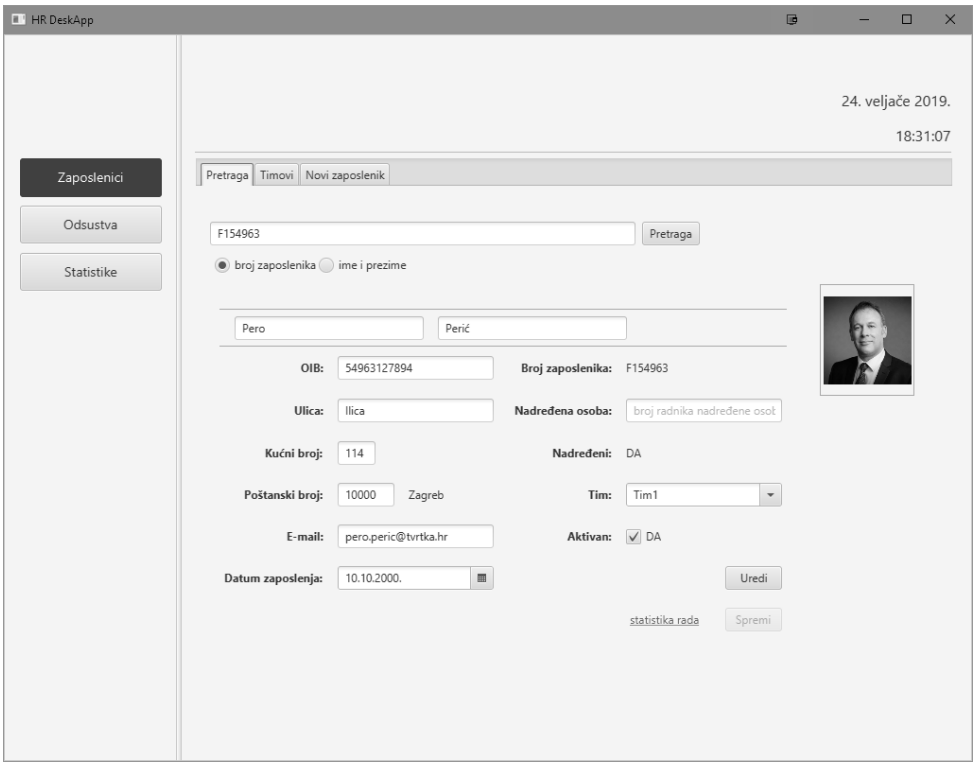


Slika 3.1. Početni ekran desktop aplikacije

Kako je prikazano na slici (Slika 3.1.), u okviru s lijeve strane nalazi se izbornik koji sadrži gumbe od kojih svaki predstavlja ulaz u pojedini modul. Na vrhu prozora aplikacije nalazi se okvir sa statusnom trakom koja prikazuje neke osnovne podatke kao što su datum, sat itd. Statusnu traku moguće je prilagoditi prema individualnim željama. Najveći i najvažniji je donji okvir u kojem se prikazuje radni dio same aplikacije. Ovisno o odabranom modulu, taj okvir mijenja svoj prikaz i funkcionalnost pa se tako u osnovnom izdanju aplikacije razlikuju sljedeći moduli: „Zaposlenici“, „Odsustva“ i „Statistike“.

3.1.1. Zaposlenici

Odabirom modula „Zaposlenici“ dohvaćaju se funkcionalnosti vezane za prikaz i ažuriranje podataka o zaposlenicima i hijerarhijskoj strukturi kompanije.

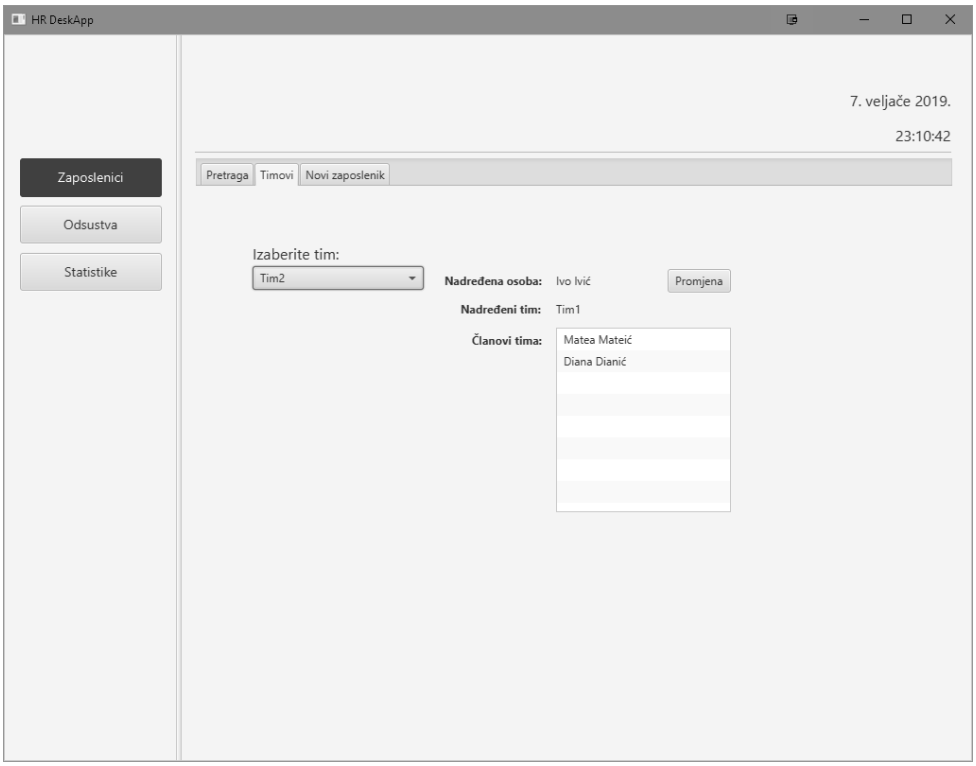


Slika 3.2. Modul "Zaposlenici", *tab* "Pretraga"

Inicijalni pogled otvara prvi po redu *tab* „Pretraga“ (Slika 3.2.) i u njemu elemente pomoću kojih se može potražiti određenog zaposlenika i dohvatiti sve njegove relevantne podatke. Pretragu je moguće izvršiti prema broju radnika ili imenu i prezimenu zaposlenika. Ukoliko se upiše jedan od potrebnih podataka koji odgovaraju sadašnjem ili bivšem zaposleniku, aplikacija će vratiti sve njegove podatke. Redom to su:

- OIB
- adresa
- e-mail adresa
- datum zaposlenja
- ime i prezime zaposleniku nadređene osobe
- podatak je li zaposlenik nadređena osobe nekome od ostalih zaposlenika
- ime tima kojem pripada
- podatak je li zaposlenik aktivan ili bivši

Također, ukoliko je zaposlenik dao dozvolu, u polju desno prikazana je slika zaposlenika. Pri dnu okvira nalazi se gumb „Uredi“. Pritiskom na taj gumb sva navedena polja postaju otključana za ažuriranje, ako za to postoji potreba, kao npr. promjena adrese stanovanja, promjena tima, prekid radnog odnosa itd. Svako polje pokriveno je validacijom specifičnom za podatak koji se u pojedino polje upisuje. Tako npr. nije moguće upisati deseteroznamenkasti OIB. Također, gumb „Spremi“ postaje aktivan te služi za konačno spremanje promjena u bazi. To sa sobom povlači i zapis *loga* s informacijama o samoj promjeni podataka. Poveznica „statistika rada“ vodi na modul „Statistike“ gdje je prikazan popis svih dolazaka i odlazaka zaposlenika čiji se podaci trenutno prikazuju, tj. upaljen je filter „prema zaposleniku“ popunjen brojem traženog zaposlenika.



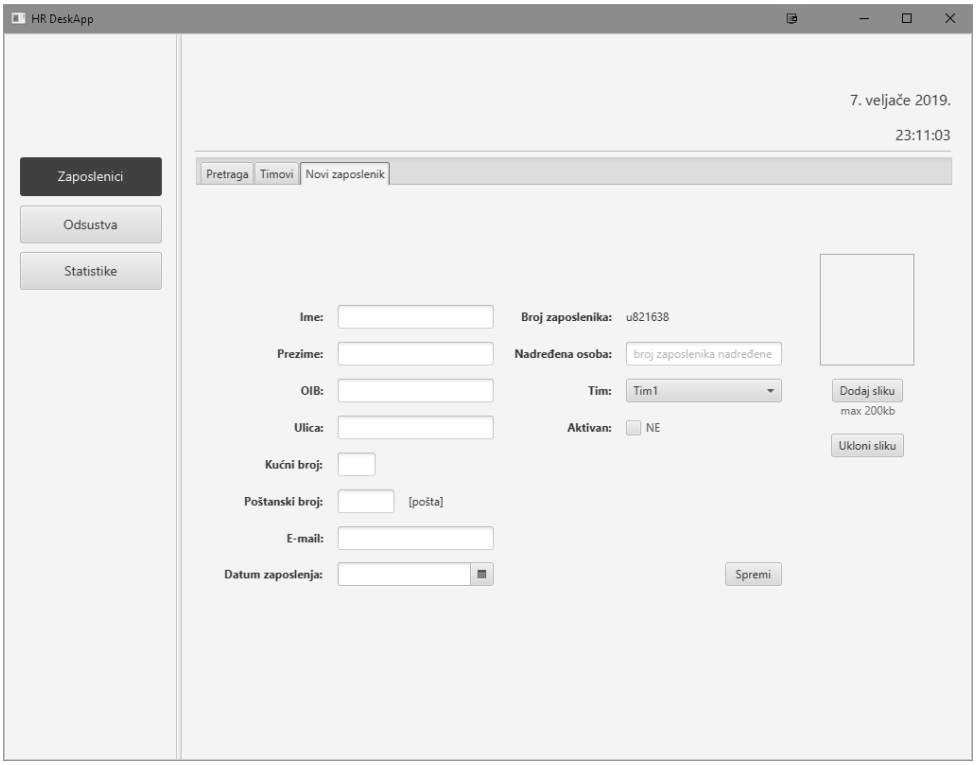
Slika 3.3. Modul "Zaposlenici", *tab* "Timovi"

Idući *tab* „Timovi“ (Slika 3.3.) prikazuje hijerarhiju kompanije u obliku popisa timova. Pomoću padajućeg izbornika može se odabrati određeni tim te se potom prikazuju podaci o tom timu. Ti podaci su:

- nadređena osoba
- nadređeni tim
- popis članova tima

Ovaj ekran dozvoljava promjenu nadređene osobe unutar tima. Ta promjena istovremeno ažurira korespondirajući podatak na razini individualnog zaposlenika. Odabirom nekog od

članova tima s popisa, ekran se prebacuje na *tab* „Pretraga“ prikazujući podatke odabranog zaposlenika te nadalje vrijede sve prethodno pobrojane funkcionalnosti tog ekrana. Bitno je napomenuti da ovdje ne postoji mogućnost manipuliranja timovima, tj. dodavanje, promjena ili brisanje timova. To je namjerno izostavljeno iz razloga što je hijerarhijski ustroj kompanije nešto što se jako rijetko mijenja i u pravilu je definirano kod osnivanja kompanije. U slučaju da do takve potrebe dođe, administratori baze mogu napraviti potrebne promjene na razini timova, a nakon toga HR pomoću aplikacije može dovršiti raspodjelu zaposlenike i druge popratne radnje.



Slika 3.4. Modul "Zaposlenici", tab "Novi zaposlenik"

Posljednji *tab* u ovom modulu je „Novi zaposlenik“ (Slika 3.4.) koji, kako mu samo ime govori, služi za dodavanje novog zaposlenika. Izgledom je veoma sličan ekranu na *tabu* „Pretraga“ kad je omogućeno uređivanje s nekoliko izmjena. Dodana su polja za unos imena i prezimena, no najbitniji element za istaknuti je „Broj zaposlenika“ koji svakim dolaskom na ovaj *tab* generira svježu šifru. To se odvija pomoću sljedeće funkcije u kôdu:

```
public String generateWorkerCode(){
        Random r=new Random();
        EmployeeSql employeeSqlImpl =new EmployeeSqlImpl();
        boolean exists=false;
        while (exists==false) {
            char c = (char) (r.nextInt('z' - 'a') + 'a');
```

```
                StringBuilder sb = new StringBuilder();
                sb.append(c);
                for (int i = 1; i <= 6; i++) {
                    Random rand = new Random();
                    int a = rand.nextInt(9) + 1;
                    sb.append(a);
                }
    if(generalBL.checkIfWorkerCodeExists(sb.toString())==false){
                    return sb.toString();
                }
            }
            return null;
        }
```

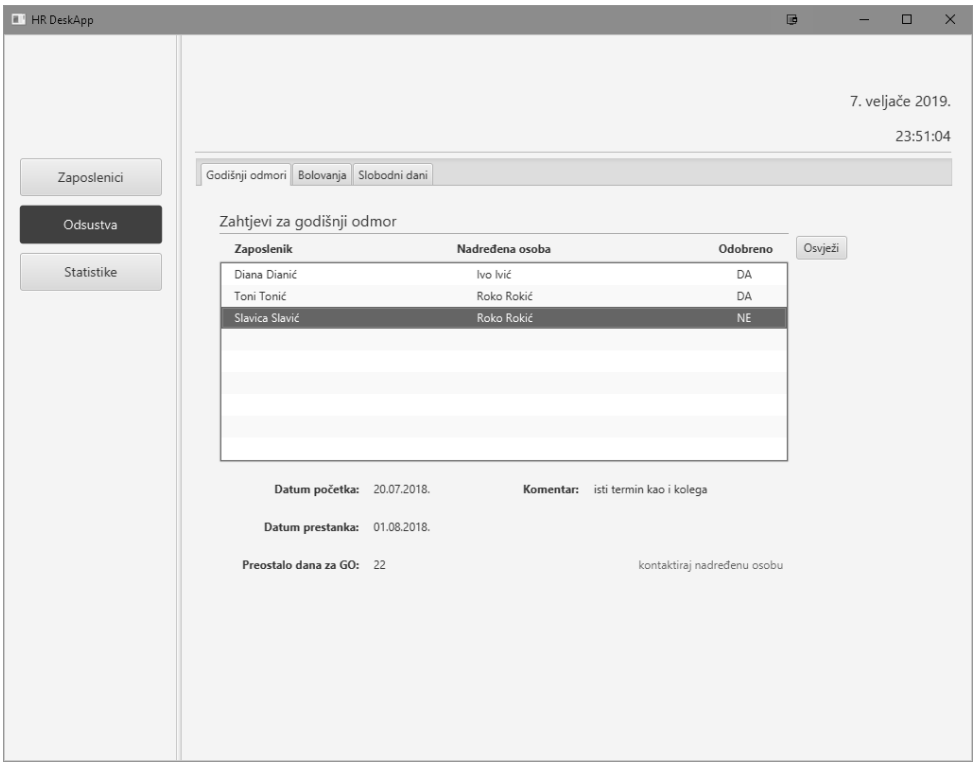
Kôd 3.1. Funkcija za generiranje novog broja zaposlenika

Ova funkcija (Kôd 3.1.) prvo generira nasumičnu šifru u formatu „*slovobrojbrojbrojbrojbrojbroj*“, a nakon toga dohvati sve zaposlenike te provjeri postoji li među njima već ta šifra. Ukoliko ne postoji, generirana šifra se prikaže na ekranu, no ukoliko postoji, prolazi istu proceduru generiranja sljedeće šifre i provjere. Na taj se način osigurava jedinstveni identitet svakog zaposlenika iz perspektive kompanije. U polju „Datum zaposlenja“ predefinirana vrijednost je uvijek tekući datum, iako je omogućena promjena iste. Time se želi sugerirati čim promptniji unos novog zaposlenika u sustav.

Odabir ekrana odrađen je pomoću *tabova* kako bi se čim više olakšalo dodavanje novih funkcionalnosti i u tom slučaju stvorio dojam uniformiranosti i konzistentnosti same aplikacije. Tako je u svim modulima.

3.1.2. Odsustva

Idući modul „Odsustva“ bavi se plaćenim i neplaćenim odsustvima zaposlenika. Prema predefiniranoj arhitekturi sva odsustva moraju biti verificirana od strane nadređene osobe zaposlenika. Zaposlenik kreira zahtjev za određeni oblik odsustva koji direktno dolazi na odobrenje nadređenoj osobi. HR je u ovom slučaju posredničko tijelo koje evidentira zahtjeve te može nadređenoj osobi eventualno proslijediti dodatne informacije. Iz tog razloga ima relativno „pasivnu“ ulogu. *Tabovi* ovog modula podijeljeni su prema tipovima odsustava pa je tako prilikom odabira modula inicijalni ekran *tab* „Godišnji odmori“ iz razloga što je to najčešći tip odsustva.



Slika 3.5. Modul "Odsustva", *tab* "Godišnji odmori"

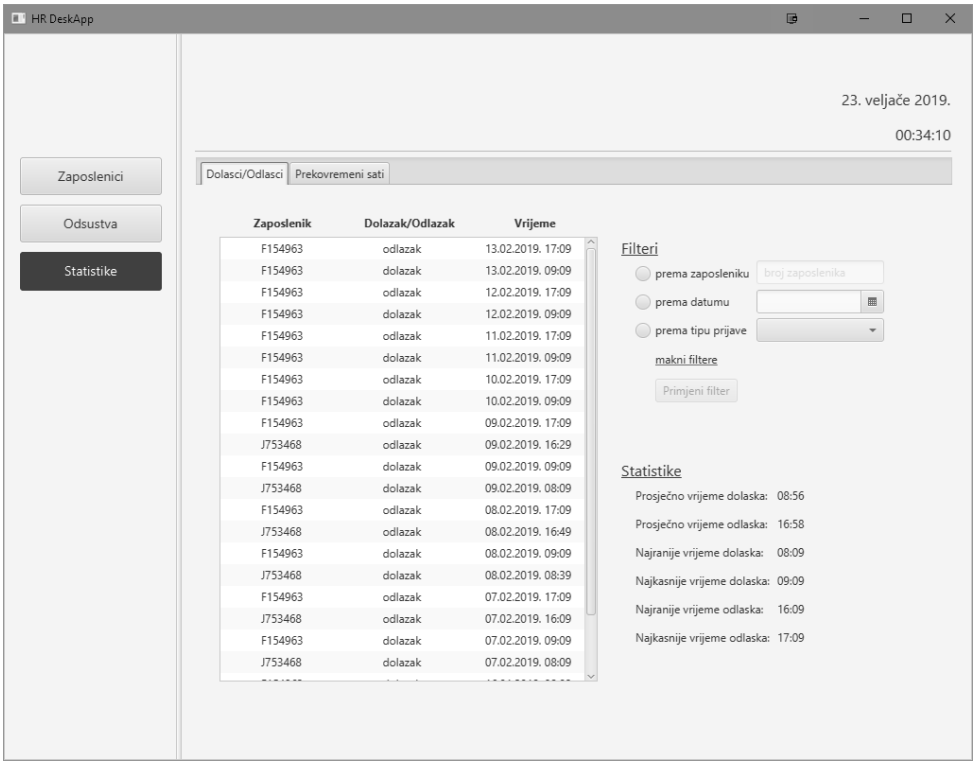
Kao što je vidljivo na slici (Slika 3.5.), aplikacija ovdje pruža mogućnost pregleda svih pristiglih zahtjeva za godišnjim odmorom te detalja pojedinačnog zahtjeva, bez mogućnosti ažuriranja ili ikakve manipulacije podacima. Nadređena osoba ima mogućnost upisati komentar na zahtjev, što je korisno ako se zahtjev iz određenog razloga odbija. Klikom na poveznicu „kontaktiraj nadređenu osobu“ otvara se postavljeni e-mail klijent s unaprijed popunjenim poljima za adresu slanja i predmet poruke. Time se smanjuje mogućnost prosljeđivanja podataka pogrešnim osobama.

Ostali *tabovi* u ovom modulu funkcioniraju na jednak način. Razlika su tip odsustva za koji se prikazuju zahtjevi te specifični elementi kao što su npr. doznake ili razlozi za opravdanje slobodnih dana. Elementi unutar ovog modula ovise o internim pravilnicima i politikama pojedine kompanije.

3.1.3. Statistike

Modul „Statistike“ i njegov inicijalni *tab* „Dolasci/Odlasci“ pruža uvid u evidenciju dolazaka i odlazaka svakog zaposlenika te prikazuje podatke koje u bazu dostavlja simulator ili implementirani realni sistem za evidenciju. Evidenciju je moguće filtrirati po više kategorija - prema zaposleniku, datumu ili tipu prijave. Navedena mogućnost HR odjelu omogućuje lakši uvid u pojedine statistike, detalje, obrasce ponašanja i slične

podatke koji mogu ilustrirati određena kretanja u samom poslovanju kompanije, a na koja je možda moguće utjecati.



Slika 3.6. Modul "Statistike", *tab* "Dolasci/Odlasci"

Ovisno o odabranom filteru, u polju „Statistike“ prikazuju se statistike prema tom filteru, npr. ako je odabran filter „prema datumu“ bit će prikazano najranije, srednje i najkasnije vrijeme dolaska zajedno s pripadajućim zaposlenikom te najranije, srednje i najkasnije vrijeme odlaska s pripadajućim zaposlenikom (Slika 3.6.).

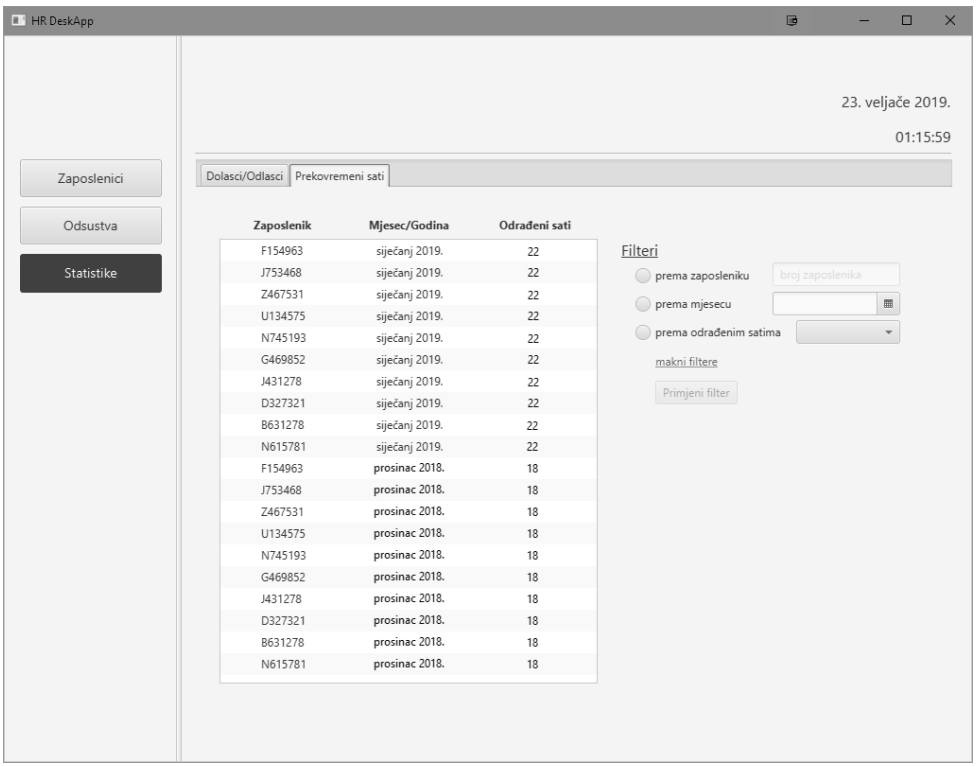
Tab „Prekovremeni sati“, kako mu ime govori, posvećen je prekovremenim satima zaposlenika (Slika 3.7.). Služi prvenstveno kako bi HR mogao krajem mjeseca dostaviti izvješće o prekovremenim satima nadležnoj računovodstvenoj službi. Također su prikazani prekovremeni sati po zaposleniku u prošlosti što, u slučaju velikog broja prekovremenih, na rukovodstvenoj razini može biti indikator određenih problema u preraspodjeli posla ili slično. Funkcija u kôdu za računanje prekovremenih sati:

```
public int getMonthlyOvertime() {
        int overtime = 0;
        ADSqlImpl adSqlImpl = new ADSqlImpl();
        double workTime =
adSqlImpl.getMonthlyWorkHoursByEmployee(ym.getMonthValue(),
ym.getYear(), this.idEmployee);
        int workTimeInt = (int) Math.ceil(workTime);
```

```
            GeneralBL generalBL = new GeneralBL();
            int totalMonthWorkTime =
(generalBL.getNumberOfWorkDaysInMonth(ym.getMonthValue(),
ym.getYear()) * 8);
            if (workTime > totalMonthWorkTime) {
                overtime = workTimeInt - totalMonthWorkTime;
            }
            return overtime;
        }
```

Kôd 3.2. Funkcija za računanje prekovremenih sati

Spomenuta funkcija (Kôd 3.2.) radi na način da pomoću evidencije dolazaka i odlazaka izračuna koliko je zaposlenik sati odradio u određenom mjesecu i to usporedi s brojem ukupnih radnih sati tog mjeseca. Funkcija je podložna promjeni u smislu ažuriranja broja obaveznih radnih sati u skladu sa zakonom i internim pravilima i politikama kompanije. Predefinirana vrijednost je 8 radnih sati dnevno, 5 radnih dana tjedno. Ista funkcija po istom principu ukazuje i na nedovoljan broj odrađenih sati te, ukoliko takva situacija nije adekvatno opravdana, može biti temelj za sazivanje disciplinskih razgovora ili sankcija zaposlenika.



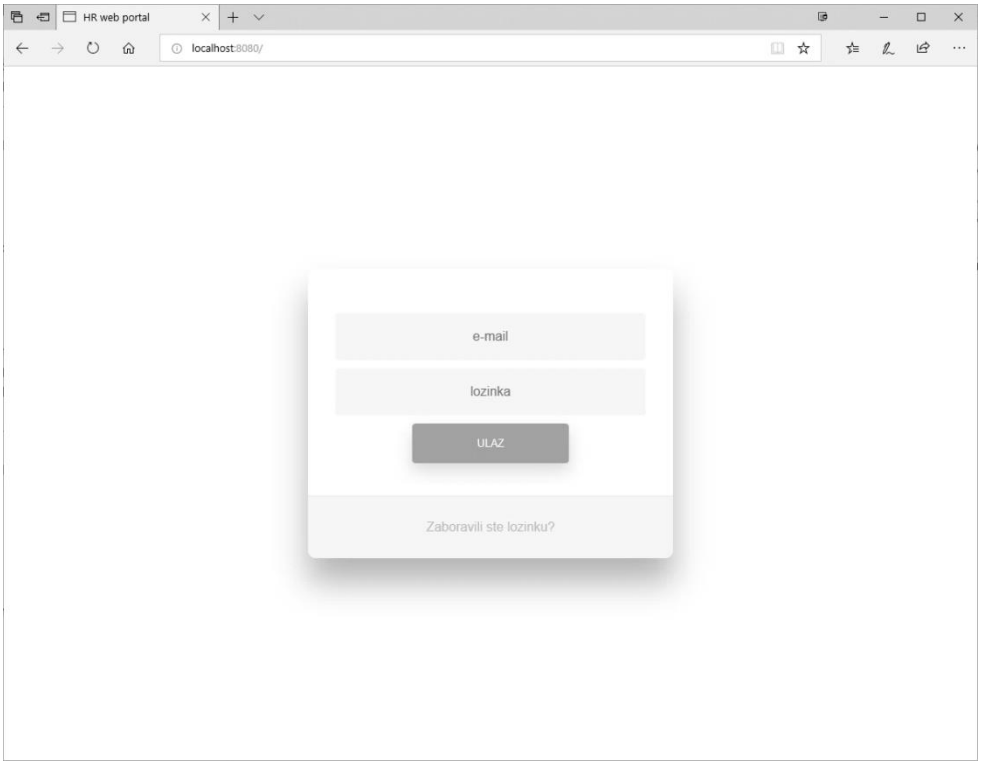
Slika 3.7. Modul "Statistike", *tab* "Prekovremeni sati"

Ovisno o eventualnim zahtjevima i potrebama pojedinog klijenta, uz dodavanje novih *tabova* u pojedine module, moguće je i dodavanje novih modula kako bi se aplikacija u potpunosti prilagodila kompaniji i njenom HR odjelu te u najvećoj mogućoj mjeri pridonijela unaprijeđenju poslovanja.

3.2. Rad u web portalu

Kao što je već spomenuto, web portalu je moguće pristupati preko bilo kojeg web preglednika u bilo koje doba.

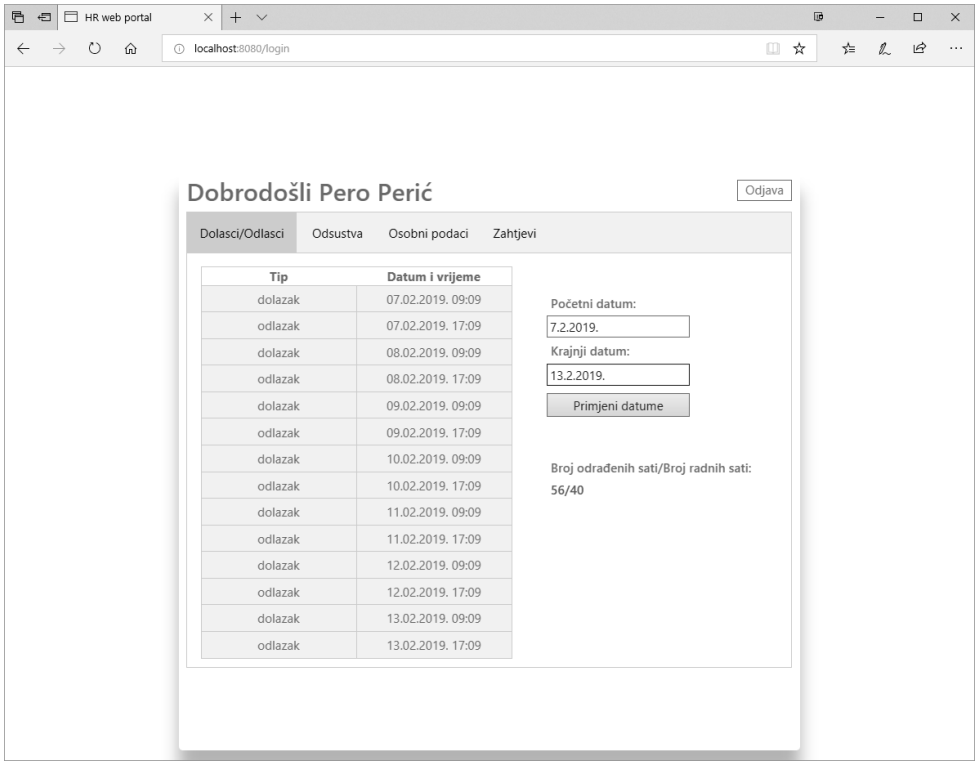
Početni ekran koji se javlja prilikom otvaranja web portala je ekran za prijavu u sam portal. Kao što je to vidljivo na slici (Slika 3.8.), sastoji se od standardnih prijavnih elemenata, polja za identifikaciju, što je u ovom slučaju službena e-mail adresa zaposlenika, te polja za upis njegove osobne lozinke. Ukoliko dođe do neuspješne prijave, pojavit će se primjerena obavijest i ponuditi mogućnost ponovnog pokušaja prijave. Po uspješnoj prijavi korisnika se prebacuje u njegov personalizirani portal koji sadrži tri ili četiri *taba*, ovisno o tome je li zaposlenik voditelj nekog od timova ili ne.



Slika 3.8. Web portal, prijavni ekran

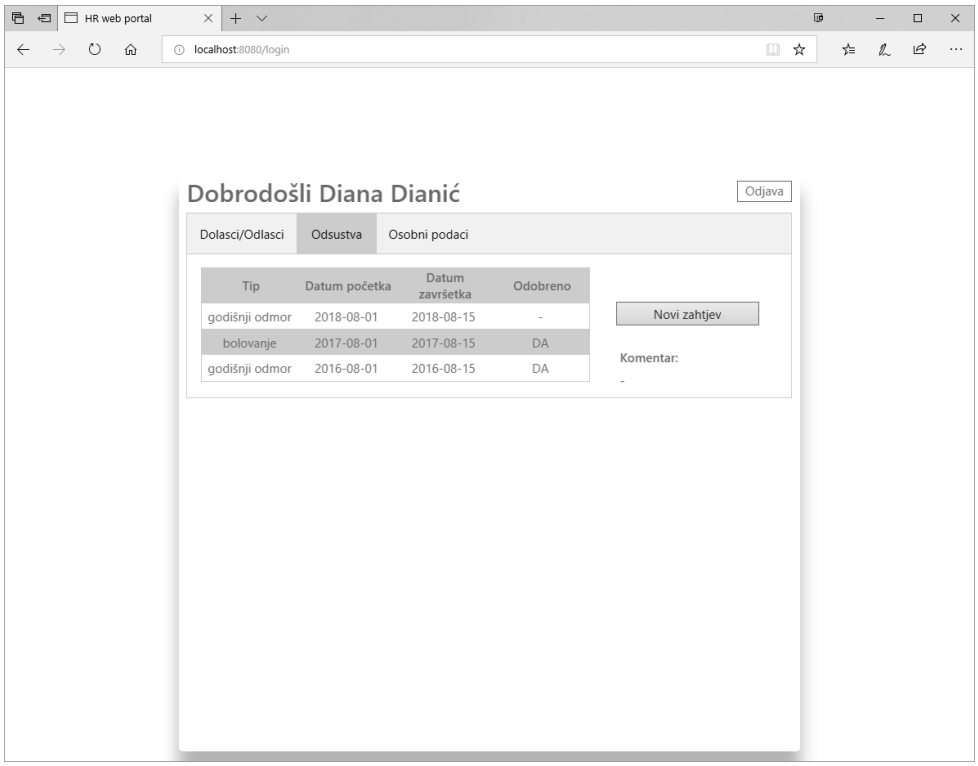
U prvom *tabu* „Dolasci/Odlasci“ nalazi se prikaz evidencije dolazaka i odlazaka. Inicijalno je prikazana evidencija za tekući mjesec (Slika 3.9.). Odabirom početnog i završnog

datuma postoji mogućnost prikaza dolazaka i odlazaka u zadanom periodu. Na ekranu se, uz evidenciju, prikazuje i agregirani broj odrađenih sati za zadani period, koji je slučaju viška ili točnog broja odrađenih sati, indikativno prikazan zelenom bojom, a u protivnom crvenom bojom. U izračun odrađenih sati naravno ulaze sva opravdana odsustva kako ne bi došlo do krivih rezultata i neugodnih situacija. Ovisno o politici kompanije, višak odrađenih sati može se kompenzirati financijski ili u obliku slobodnih dana. U slučaju da zaposlenik odabere slobodne dane kao kompenzaciju, na HR odjelu je da te dane unese u evidenciju kako bi svi parametri na kraju tog perioda odgovarali planiranom.



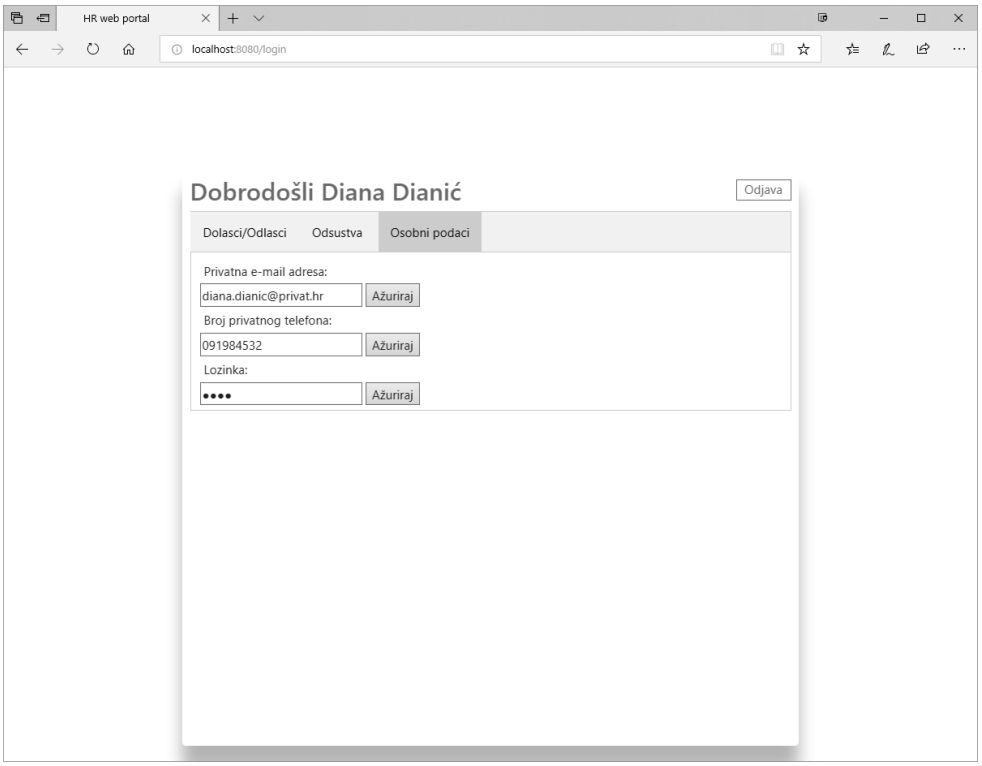
Slika 3.9. Web portal, *tab* "Dolasci/Odlasci"

Sljedeći *tab* „Odsustva“ služi za pregled postojećih i formiranje novih zahtjeva za odsustvima. U polju za prikaz zahtjeva vidljivi su svi aktualni i prijašnji zahtjevi za odsustvima zaposlenika te njihove osnovne informacije, kao što je prikazano na slici (Slika 3.10.). Odabirom nekog od zahtjeva, s desne strane se prikazuju svi detalji tog zahtjeva. Klikom na gumb „Novi zahtjev“ otvara se forma u kojoj se trebaju popuniti potrebni podaci kako bi se kreirao novi zahtjev za odsustvom. Po uspješnom kreiranju, novi zahtjev je odmah vidljiv u polju za prikaz zahtjeva te čeka očitovanje voditelja zaposlenika.



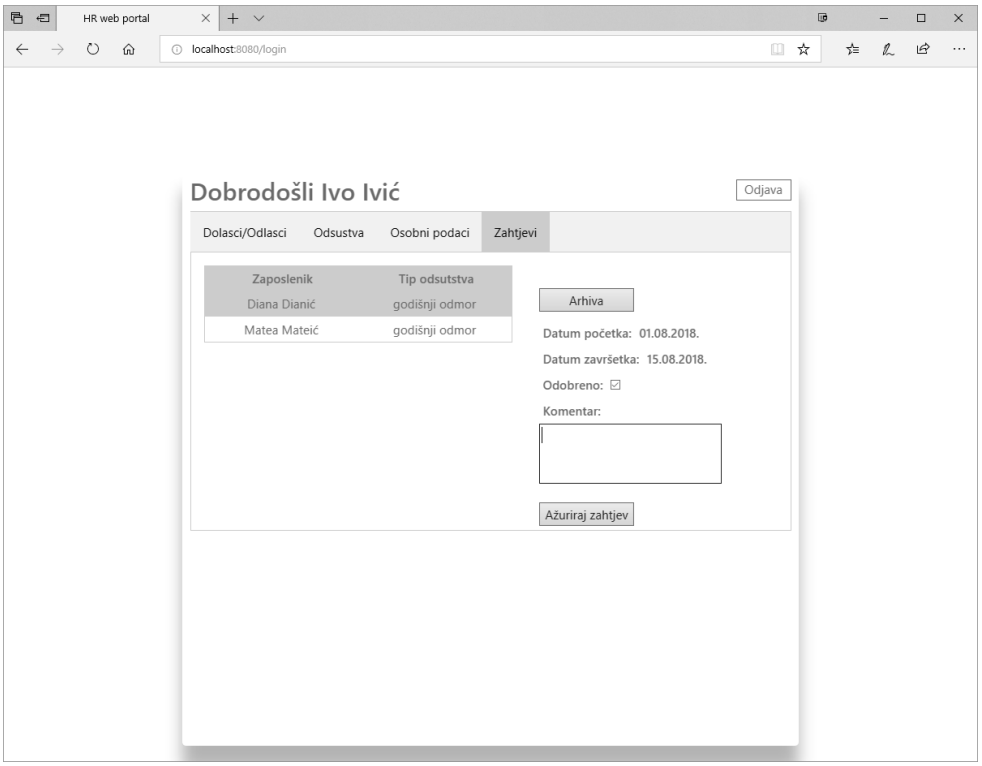
Slika 3.10. Web portal, *tab* "Odsustva"

Tab „Osobni podaci“ služi za promjenu i ažuriranje nekih osobnih podataka zaposlenika kao što su privatna e-mail adresa, privatni broj telefona, lozinka za pristup web portalu itd. (Slika 3.11.). Promjena ovih podataka, osim lozinke, odmah je vidljiva i kroz desktop aplikaciju u HR odjelu. Iz sigurnosnih razloga promjena lozinke zahtjeva ponovni upis trenutne lozinke i dvostruki upis nove. Svrha ovog ekrana je pružanje zaposlenicima mogućnosti kontrole nad svojim podacima, ali prvenstveno da eliminira potrebu intervencije HR odjela u svrhu promjene tih podataka te time smanji broj procedura u poslovanju rasterećujući kako zaposlenika, tako i HR. U svakoj kompaniji postoji velik broj ovakvih malih zahvata koji često zahtijevaju pažljivu obradu i povećavaju opterećenje na nadležno ljudstvo te se njihovom eliminacijom oslobađa vrijeme i prostor za obradu drugih stvari koje su važnije za poslovanje.



Slika 3.11. Web portal, *tab* "Osobni podaci"

Kao što je već napomenuto, *tab* „Zahtjevi“ vidljiv je samo zaposlenicima koji su voditelji nekih od timova. U njemu su vidljivi zahtjevi za odsustvima podređenih zaposlenika koji čekaju očitovanje, tj. odobrenje ili odbijanje zahtjeva (Slika 3.12.). Voditelj ima mogućnost dodati komentar na zahtjev kojim obrazlaže svoju odluku. Potvrdom odluke o zahtjevu, taj zahtjev nestaje iz prikaza aktualnih zahtjeva te odlazi u arhivu. Također, nakon potvrde, zaposlenika koji je vlasnik zahtjeva obavještava se e-mailom o promjeni statusa njegovog zahtjeva. Klikom na gumb „Arhiva“ mijenja se prikaz zahtjeva tako da se prikazuju svi prošli zahtjevi oko kojih se voditelj već očitovao. Zahtjeve iz arhive nije moguće mijenjati ni u kom smislu, već taj prikaz služi isključivo za svrhu revizije ili slične situacije u kojima treba dokaz ili opravdanje pojedinog zahtjeva ili odluke. Ukoliko se ipak pokaže potreba za izmjenom nekog zahtjeva iz arhive, voditelj mora kontaktirati HR koji će odlučiti o daljnjim koracima u skladu s politikama kompanije.



Slika 3.12. Web portal, *tab* "Zahtjevi"

3.3. Prednosti rada u aplikaciji

Kako je vidljivo iz prikaza svih ekrana, web portal je u suštini sažetija verzija desktop aplikacije koju koristi HR odjel, fokusirana na individualne podatke. To dovoljno ilustrira tendenciju ovog sustava da HR odjelu maksimalno olakša prikupljanje podataka i obavljanje zadataka uključujući pritom svakog zaposlenika, istovremeno ubrzavajući velik broj procesa i procedura vezanih za HR. Također, osim nezaobilaznih ugovora i ostalih mandatornih fizičkih dokumenata, u potpunosti se anulira potreba za papirnatom dokumentacijom.

Jedna od prednosti je dakako i već spomenuta jednostavna mogućnost nadogradnje aplikacije. Nadogradnje su posljedice želja i potreba individualnih klijenata. Primjer moguće nadogradnje koja bi u većim kompanijama zasigurno polučila kvalitetne i značajne rezultate i uštede je dodavanje modula za kreiranje *custom* sadržaja u desktop aplikaciju. To bi moglo poslužiti primjerice, za kreiranje anketa, upitnika ili obaveznih edukacija koje svi zaposlenici moraju proći. Dodavanjem dodatnog *taba* u web portal u kojem bi se prikazali *custom* sadržaji kreirani od strane HR odjela, u potpunosti se digitalizira odnos između zaposlenika i HR odjela kompanije. Kako se u većim kompanijama za provođenje

anketa ili statističkih upitnika zbog opterećenosti internog HR odjela često angažiraju vanjski partneri, implementacija ovog sustava i spomenutih nadogradnji može dovesti i do direktnih financijskih ušteda.

S druge strane, neovisno o veličini kompanije, sustav pruža jednolik i sadržajan uvid u statistiku i produktivnost ljudskih potencijala. Čak i minimalne korekcije i intervencije, kao posljedice uvida u te podatke, mogu dovesti do značajnog unaprijeđenja efikasnosti i kvalitete poslovanja što je konačni cilj svake kompanije.

Zaključak

U poslovanju danas veliki naglasak stoji na upravljanju ljudskim potencijalima jer su zaposlenici ti koji donose kvalitetu i napredak te ih konstantno treba poticati i nadograđivati u pravom smjeru. Da bi se to postiglo, mogućnost pregleda i kontrole nad svim relevantnim informacijama s jednog mjesta ima nemjerljivu vrijednost.

Aplikacija koja je tema ovog rada, prijedlog je koraka u tom smjeru jer objedinjuje pristup i manipulaciju podacima o zaposlenicima pomoću desktop aplikacije u za to zaduženom HR odjelu, kao i pojedinačni pristup osobnim podacima svakom zaposleniku individualno putem web portala. Uz to integrira i sustav evidencije dolazaka i odlazaka koji daje iznimno realnu sliku produktivnosti kompanije koja je implementirala rješenje.

Korištenjem svih dijelova aplikacije uvelike se olakšava upravljanje ljudskim potencijalima i smanjuju obujmi HR procedura koji su često opširni i tromi te dodatno opterećuju poslovanje. Vizualno „lagan“ i intuitivan dizajn omogućuje instantnu implementaciju rješenja bez potrebe za dodatnim edukacijama korisnika. Rješenje također pruža statistike i podatke korisne za ostale odjele unutar kompanije.

Generička priroda, jednostavna implementacija i mogućnost lagane nadogradnje najveći su aduti ovog rješenja zbog činjenice da ne podrazumijeva veliki angažman klijenta, a pruža mu cjelovit i pouzdani sustav koji direktno ili indirektno može donijeti pozitivne pomake u poslovanju.

Popis kratica

CSS	Cascading Style Sheets	kaskadni stilovi
HR	Human Resources	ljudski potencijali
HTML	HyperText Markup Language	hipertekstualni označni jezik
HTTP	Hyper Text Transfer Protocol	protokol za prijenos hiperteksta
MVC	Model View Controller	model pogled upravljač
ODBC	Open DataBase Connectivity	povezivost s otvorenom bazom podataka
SQL	Structured Query Language	jezik strukturiranih upita
XML	eXtensible Markup Language	proširivi označni jezik

Popis slika

Slika 2.1. *Shema* arhitekture sustava 5

Slika 2.2. Simulator evidencije dolazaka i odlazaka zaposlenika 11

Slika 3.1. Početni ekran desktop aplikacije 12

Slika 3.2. Modul "Zaposlenici", *tab* "Pretraga" 13

Slika 3.3. Modul "Zaposlenici", *tab* "Timovi" 14

Slika 3.4. Modul "Zaposlenici", tab "Novi zaposlenik" 15

Slika 3.5. Modul "Odsustva", *tab* "Godišnji odmori" 17

Slika 3.6. Modul "Statistike", *tab* "Dolasci/Odlasci" 18

Slika 3.7. Modul "Statistike", *tab* "Prekovremeni sati" 19

Slika 3.8. Web portal, prijavni ekran 20

Slika 3.9. Web portal, *tab* "Dolasci/Odlasci" 21

Slika 3.10. Web portal, *tab* "Odsustva" 22

Slika 3.11. Web portal, *tab* "Osobni podaci" 23

Slika 3.12. Web portal, *tab* "Zahtjevi" 24

Popis kôdova

Kôd 2.1. *Singleton pattern* za pristup bazi podataka iz aplikacije 4

Kôd 2.2. Primjer funkcije pisane u JavaScript programskom jeziku 8

Kôd 2.3. Primjer petlje korištenjem *scriptleta* u JSP datoteci..... 10

Kôd 2.4. Primjer *if* bloka korištenjem JSTL *Core* biblioteke..... 10

Kôd 3.1. Funkcija za generiranje novog broja zaposlenika..... 16

Kôd 3.2. Funkcija za računanje prekovremenih sati 19

Literatura

[1] WIKIPEDIA, Java (programing language), https://en.wikipedia.org/wiki/Java_(programming_language), 3. veljače 2019.

[2] ORACLE CORPORATION, What is Java, https://www.java.com/en/download/faq/whatis_java.xml, 22. veljače 2019.

[3] TECHTARGET, Microsoft SQL Server, https://searchsqlserver.techtarget.com/definition/SQL-Server, 22. veljače 2019.

[4] ORACLE, Java Documentation, https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm, 20. veljače 2019.

[5] SPRING, Spring Framework, https://docs.spring.io/spring/docs/current/spring-framework-reference/overview.html, 22. veljače 2019.

[6] SPRING, https://spring.io, 15. veljače 2019.

[7] SPRING, Web on Servlet Stack, https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html, 15. veljače 2019.

[8] MDN WEB DOCS, About JavaScript, https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript, 22. veljače 2019.

[9] WIKIPEDIA, Bootstrap (front-end framework), https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework), 15. veljače 2019.

[10] INTRODUCTION – BOOTSTRAP, https://getbootstrap.com/docs/4.3/getting-started/introduction/, 15. veljače 2019.

[11] APACHE TOMCAT, http://tomcat.apache.org/, 22. veljače 2019.

[12] THESERVERSIDE, Apache Tomcat, https://www.theserverside.com/definition/Tomcat, 22. veljače 2019.

[13] WIKIPEDIA, Singleton pattern, https://en.wikipedia.org/wiki/Singleton_pattern, 19. veljače 2019.

[14] TOPTAL, Single responsibility principle: A recipe for great code, https://www.toptal.com/software/single-responsibility-principle, 22. veljače 2019.

[15] WIKIPEDIA, JavaServer Pages, https://en.wikipedia.org/wiki/JavaServer_Pages, 21. veljače 2019.

[16] KUČAK, D. *Izrada web aplikacija (priručnik)*. Algebra, Zagreb, 2013.

[17] TUTORIALSPOINT, JSP – Standard Tag Library (JSTL) Tutorial, https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm, 21. veljače 2019.

ALGEBRA
VISOKO UČILIŠTE

APLIKACIJA ZA POMOĆ U RADU S LJUDSKIM POTENCIJALIMA

Pristupnik: Marko Cmrk, 0054026345

Mentor: prof. Aleksander Radovan