

Dr. sc. Nevzudin Buzadija
Lejla Omerašević

AHA! – APLIKACIJA ZA UČENJE

Sažetak

U radu će se analizirati na koji način možemo da poboljšamo motivaciju kod učenika uz pomoć integrativne nastave i primjene softverskih rješenja. Također, rad će prikazati faze projektovanja i razvoja aplikacije u Java programskom okruženju namjenjenu učenicima i profesorima, a koja je namjenjena učenju raznih sadržaja i testiranja učenika. Prilikom razvoja aplikacije koristila se metodologija koja omogućava siguran put do rješenja i olakšava pisanje koda. Aplikacija je jednostavna za korištenje i ne traži napredno poznavanje rada na računaru što će stvarati manji otpor kod učenika i profesora da prihvate ovu aplikaciju.

Ključne riječi: Java, modeliranje, motivacija, kviz, lekcije, izvorni kod, UML dijagrami

Uvod

Na učenje kao neprekidan cjeloživotni proces potiču svakodnevne aktivnosti koje prati trenutni, savremeni svijet. Učenje počinje od prvog dana našeg života, gdje su glavni učitelji roditelji. Nakon toga slijedi konkretno učenje kroz nastavni proces u školskim ustanovama, gdje ulogu učitelja preuzimaju školovani, stručni kadar: učitelji, nastavnici, profesori. Nastavnicima u Bosni i Hercegovini je oko 10 sati sedmično plaćeno za pripreme prilikom rada sa učenicima. Prosječan broj učenika kojem jedan nastavnik predaje je 250, što je oko 2,5 minuta po učeniku. Upravo zbog toga nastavnici nerijetko i utrostruče taj broj sati radi kvaliteta nastave. Ako dodamo na to pedagošku dokumentaciju, vrijeme ispravljanja kontrolnih radova, izdavanje sopstvenog novca za nastavni materijal itd., očigledan je zaključak da je nastavnikov posao izuzetno zahtjevan.

Današnja *net generacija* učenika i nastavnika uglavnom smatra da je ovakav sistem obrazovanja uveliko zastario. Naglasak je na tome da se treba učenik naučiti kako da misli, da zaključuje,

analizira, te kako pristupiti nekom problemu na njima interesantan način.

Upravo zbog toga je sve više aplikacija za učenje koje se koriste u nastavi. U ovom radu ćemo prezentovati aplikaciju za učenje pod nazivom „AHA!“ u kojoj učenik usvaja i testira usvojena znanja na njemu zanimljiv način. Prikazati ćemo sve faze kreiranja jednog sistema, kao i njegove karakteristike.

Dosadašnja istraživanja

Respektivne edukacijske ustanove i studenti širom svijeta su tokom 2011. godine postavili pitanje povodom porasta broja mnogobrojnih online kurseva i usvajanja znanja pomoću aplikacija. Takve aplikacije uključuju lekcije u formatu tekstova i videa kombinovanih sa kvizovima za provjeru znanja. Izvršilo se poređenje navedenih sistema za učenje sa učenjem „licem u lice“. Iako je poprilično teško pronaći povratnu informaciju za efikasnost ovakvog sistema, istraživanje je ipak uspjelo pronaći dokaze koje sugerišu da nema razloga smatrati da je učenje „licem u lice“ efikasnije od online učenja pomoću aplikacija. (David, G. G., Martin, F., Myles, R., 2013)

Park univerzitet iz Chicaga je izdao članak pod nazivom „A Faculty Observation Model for Online Instructors: Observing Faculty Members in the Online Classroom“ gdje su se, uz korištenje naprednog učenja, posmatrali efekti istog. U članku su dati statistički podaci povratnih informacija studenata o efikasnosti obrazovanja pomoću online sistema. Globalno gledajući, 75% učenika smatra da ovakav pristup učenju ispunjava, ili premašuje njihova očekivanja. Univerzitet smatra da se mora svakodnevno unapređivati i održavati takav sistem učenja, prilagođavati se potrebama učenika i nastavnika, te upošljavati stručni kadar za realizaciju svega navedenog. (Michael, T.E., Henry, R., 2013)

Članak iz „Journal of the Medical Library Association : JMLA“ daje zaključke pri korištenju anonimnih pop-kvizova kao alata za poboljšanje učenja. Urađeno je istraživanje sa učenicima, gdje su ukupno postigli rezultate od 79% tačnosti. Istraživanje je pokazalo da sudjelovanje u kvizu odmah nakon usvajanja novog znanja je mnogo efektivnije u odnosu na samostalno učenje bez kviza. (ncbi, 2012)

Kratak opis projekta

Zadatak projekta glasi: Napisati Java program koji implementira aplikaciju za savladavanje gradiva za učenike pod nazivom „AHA!“. Kreirati prozor za logovanje i registraciju učenika i profesora.

Dio za učenike:

1. Učenici biraju nastavnu jedinicu čija znanja žele da usvoje, ili utvrde,
2. Nastavnu jedinicu podijeliti na deset dijelova, na kojima će biti teoretski dio kojeg treba savladati, a zatim učenik kroz pitanja tipa „odaberi tačan odgovor“ provjerava svoje usvojeno znanje.
3. Svaki odgovor bodovati jednim bodom i na kraju savladavanja nastavne jedinice prikazati učeniku koliko je bodova osvojio, te prikazati tačne odgovore.

Dio za profesore:

1. Prikaz osvojenih bodova svakog učenika,
2. Prikaz iz koliko je pokušaja učenik savladao znanja,
3. Vremenski prikaz učenikovog pristupanja testu.
4. Mogućnost kreiranja lekcije i pitanja.

Cilj projekta je kreiranje aplikacije koja će učenicima na zanimljiv način prezentovati lekcije i omogućiti testiranje iz prezentovanog gradiva.

Analiza zahtjeva

U ovoj fazi se u saradnji sa kupcem utvrđuju zahtjevi koji opisuju sistem. Pri njihovom definisanju uzimaju se u obzir entiteti, aktivnosti i ograničenja koja postoje u sistemu, kao i interakcija sistema sa okruženjem.

Zahtjevi kupca „AHA!“ projekta su:

1. Kreirati sistem koji će omogućavati samostalno usvajanje i testiranje znanja za svrhe usvajanja nove nastavne jedinice, utvrđivanja gradiva, ili zadavanje domaće zadaće,
2. Sistem treba imati prepoznatljivo ime i logo,
3. Treba biti lagan za upotrebu, prilagođen korisniku,

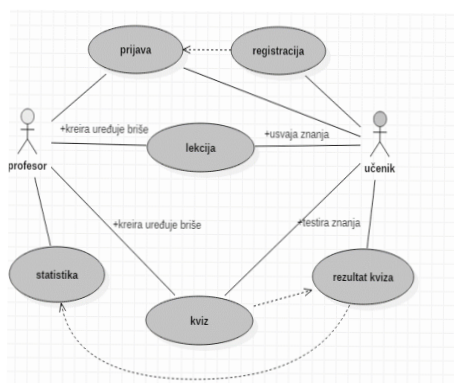
4. Dizajn treba biti jednostavan,
5. Softver treba biti višekorisnički sa dodijeljenim ulogama i ograničenim pristupom,
6. Omogućiti logovanje na sistem,
7. Omogućiti profesoru kreiranje, izmjenu i brisanje teoretskog dijela za usvajanje nastavne jedinice, kreiranje, izmjenu i brisanje pitanja kviza od deset pitanja na bazi „odaberi tačan odgovor“,
8. Omogućiti učeniku registraciju, pregledanje lekcije, odgovaranje na pitanje u kvizu,
9. Nakon kviza, učeniku omogućiti uvid u bodove i tačne odgovore, kao i spremanje bodova
10. Omogućiti profesoru da ima uvid u bodove učenika, vrijeme učenikovog pristupanja kvizu i broj spremljenih pokušaja,
11. Program ne smije imati veliku hardversku zavisnost,
12. Instalacija softvera mora biti jednostavna,
13. Softver treba biti podržan za sve operativne sisteme,
14. Navesti koju softversku podlogu za instaliranje i upotrebu softvera računar mora posjedovati.

Kritične tačke (milestones):

Tabela 1. Vremenski period ispunjavanja kritičnih tačaka

ZAHTJEV	VREMENSKI PERIOD
Utvrdjivanje imena i kreiranje logo-a sistema	15 dana
Interfejs lekcije	15 dana
Interfejs kviza	15 dana
Interfejs rezultata kviza	30 dana
Korisničko uputstvo	30 dana

Dijagram slučajeva korištenja (use-case diagram) bi izgledao ovako:



Slika1. Prijedlog Use Case dijagrama

Projektovanje sistema

Projektovanje sistema je kreativni, interaktivni proces prevođenja problema u njegovo rješenje, koristeći specifikaciju zahtjeva iz prethodne faze. U tom procesu bitno je kreirati konceptualni i tehnički dizajn. Konceptualni dizajn odgovara na pitanje šta sistem radi, identifikuje sve entitete, attribute i njihove veze, daje korisnički interfejs, opisuje vremenski tok događaja u sistemu, te daje informacije o ulaznim i izlaznim podacima. Tehnički dizajn odgovara na pitanje kako sistem radi. Namijenjen je razvojnom timu, te utvrđuje konfiguraciju računarskog sistema za realizaciju, daje način komunikacije u sistemu, definiše ulaze i izlaze sistema i opisuje arhitekturu sistema.

Konceptualni dizajn za „AHA!“ projekat

Sistem koji će omogućavati samostalno usvajanje i testiranje znanja za svrhe usvajanja nove nastavne jedinice, utvrđivanja gradiva, ili zadavanje domaće zadaće će biti pod nazivom „AHA!“. Naziv aplikacije je prepoznatljiv i koristit će se kao dugme za „dalje“ kod usvajanja nove nastavne jedinice. Logo izgleda ovako:



Slika 2. Logo aplikacije

Pri pokretanju programa pojavljuje se pozor za prijavu/registraciju. Sistem je napravljen na principu klijent – server, tako da imamo projektnog administratora (kompanija koja kreira sistem), sistemskog administratora (profesora) i učenike kao klijente.

Učenik nema zadano korisničko ime, te se prvo mora registrovati pomoću dugmeta „REGISTRACIJA“. U prozoru „Registracija“ potrebno je unijeti ime i prezime, šifru, e-mail, razred i odjeljenje, a zatim odabrati dugme „Prijava“, nakon kojeg se otvara dijaloški okvir koji prikazuje poruku „Unos podataka je uspješan“. Pritiskom na dugme „OK“ iz dijaloškog okvira, sistem nas vraća na početni prozor za prijavu.



Slika 3. Prozor za prijavu



Slika 4. Prozor za registraciju

Svi podaci registrovanih učenika se pohranjuju u bazu podataka, tako da je omogućeno svako dalje prijavljivanje sa istim podacima.

Sistemsom administratoru nakon logovanja se otvara prozor sa sljedećim interfejsom:

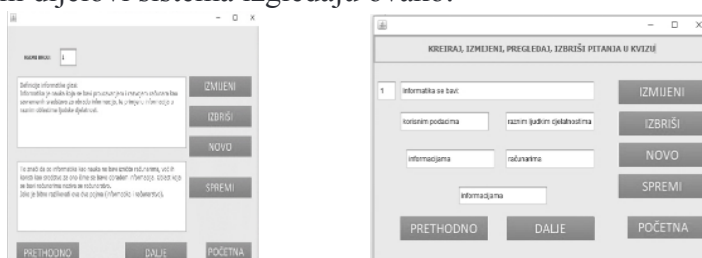


Slika 5. Početna – profesor

Unutar prozora nalaze se dugmad koja imaju sljedeće osobine:

- Lekcija – otvara prozor koji sadrži tri polja za redni broj i tekstualne podatke koji se mogu pisati, uređivati i brisati, te dugme „AHA!“ pomoću kojeg se otvaraju dva nova tekstualna polja. Profesor može kreirati neograničen broj polja da bi što bolje učenicima predočio znanja iz određene nastavne jedinice.
- Kviz – analogno lekciji, profesor ima istu ulogu u kreiranju pitanja kviza koji se sastoji od 4 moguća odgovora, te tačnog odgovora. Kada se odabere dugme „spremi“, prikazuje se prvo pitanje.
- Statistika – profesor ima uvid u tabelarni prikaz učenikovih bodova koje je osvojio pristupajući kvizu, uvid u datum, vrijeme i broj pokušaja učenika da eventualno osvoji maksimalan broj bodova.

Navedeni dijelovi sistema izgledaju ovako:



Slika 6. Prozori za kreiranje lekcije (slika lijevo) i kreiranje kviza

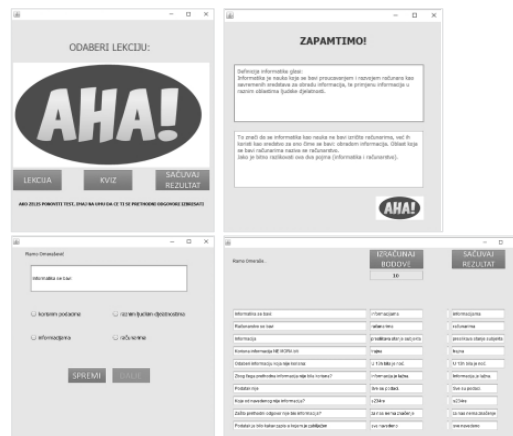


Slika 7. Prozor statistike

U trenutku prijave učenika otvara se glavni prozor koji nudi izbor pomoću dugmadi:

1. Lekcija – otvara prozor nastavne jedinice koju je kreirao profesor za učenike. Dugme „AHA!“ ima osobinu „dalje“.
2. Kviz – otvara prvo pitanje kviza kojeg je kreirao profesor. Kviz se sastoji od 10 pitanja i samo je jedan tačan odgovor. Odabir odgovora je kreiran na način da se jedan od ponuđenih mora izabrati, zatim spremi odabir, te ići dalje. U glavnom prozoru je poruka učenicima koja glasi: „Ako želiš ponoviti test, imaj na umu da će ti se prethodni odgovori izbrisati!“.
3. Sačuvaj rezultat – nakon izrade kviza učeniku se ostavlja opcija da ima uvid u tačne odgovore, u lični rezultat, te da, ukoliko želi, da spremi ostvareni uspjeh, pošto je cilj aplikacije da učenik usvoji, ili utvrdi znanja, a ne evaluaciju istih.

Navedeni prozori izgledaju ovako:



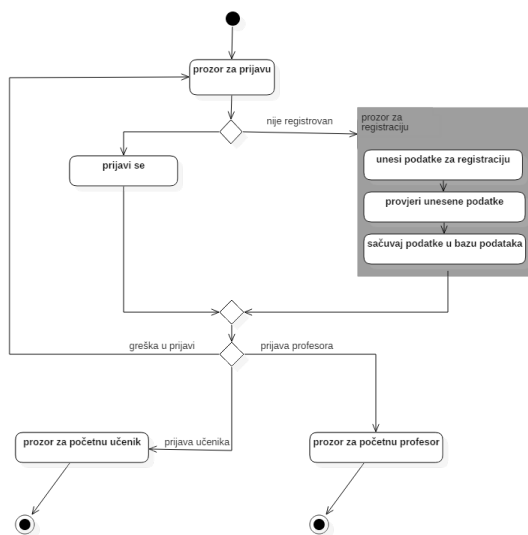
Slika 8. Prikazi dijelova aplikacije za učenike

UML modeliranje

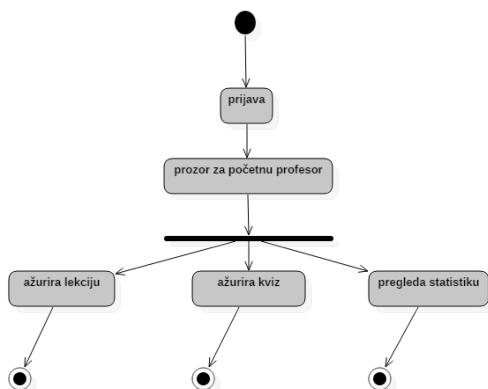
Modeli softverskog inženjerstva opisuju proces razvoja softverskog sistema. Jedan od načina grafičkog prikazivanja softverskog inženjerstva jeste pomoću jezika za modeliranje pod nazivom UML koji nam omogućava prikaz funkcionalnosti i arhitekture sistema pomoću raznih dijagrama. U nastavku slijedi prikaz zadanog sistema pomoću UML dijagrama.

Dijagram aktivnosti

Ovaj dijagram predstavlja kombinaciju aktivnost u sistemu. Aktivnost je funkcija koju sistem izvodi. Dijagrami aktivnosti se koriste za prikazivanje sveobuhvatne kontrole toka akcije u sistemu. Za ovaj projekat su kreirana tri aktivna dijagrama i to: dijagram za prijavu i registraciju, dijagram aktivnosti profesora i dijagram aktivnosti za pregled pitanja iz kviza (aktivnost profesora):

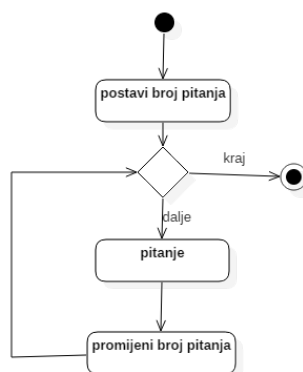


Slika 12. Dijagram aktivnosti za prijavu i registraciju



Slika 13.

Dijagram aktivnosti
za profesora

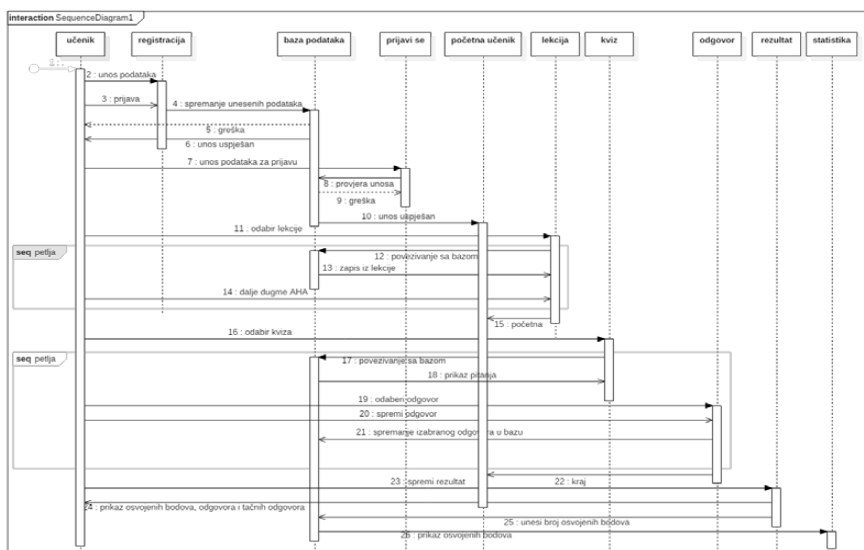


Slika 14.

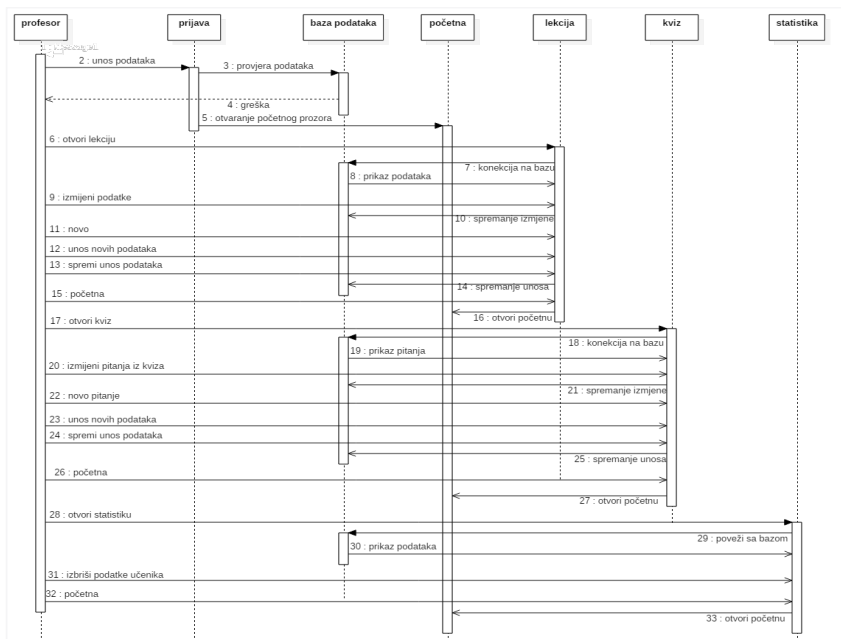
Dijagram aktivnosti za
pregled pitanja u kvizu
(profesor)

Dijagrami sekvence

Dijagram sekvence je interaktivni dijagram koji pokazuje kako procesi funkcioniraju jedan sa drugim i kojim redom. Takva komunikacija je prikazana paralelnim vertikalnim linijama koje nazivamo linije života. Horizontalne linije prikazuju razmjenu poruka između procesa onim redom ako su se desile.



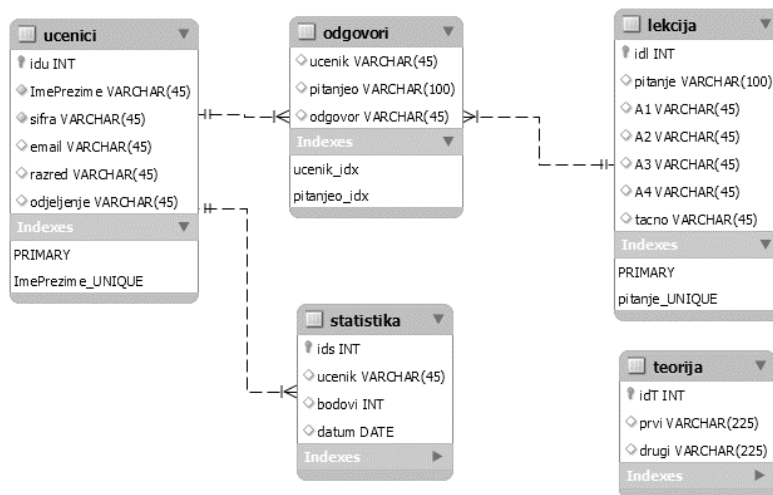
Slika 15. Sekvencijalni dijagram sa stanovišta učenika kao aktera



Slika 16. Sekvencijalni dijagram sa stanovišta učenika kao aktera

Projekat „AHA!“ će se kreirati u programskom jeziku Java uz pomoć softvera Netbeans. Tako će se sve definisane klase implementirati zajedno sa modelima, interfejsom i samom arhitekturom sistema. Uz Java programiranje, potrebno je kreirati i bazu podataka na standardnom jeziku za skladištenje podataka i u tu svrhu će se koristiti MySQL server.

Sam izgled baze podataka izgleda ovako:



Slika 18. E-R dijagram aplikacije

Implementacija softvera

U ovom dijelu ćemo prikazati dio karakterističnih dijelova koda pisanih u Java programskom jeziku zbog nemogućnosti da se prikaže cijeli kod kojeg ima cca 100 stranica A4 formata.

Prijava – naziv klase je login:

```
import java.awt.Toolkit; /*korištenje konkretne klase*/ import
java.sql.*; /*korištenje svih klasa iz paketa*/ import javax.swing.
JOptionPane; import java.awt.event.*; import java.awt.*; public
class login extends javax.swing.JFrame { /*deklaracija klase login
koja predstavlja podklasu roditeljske klase javax.swing.JFrame*/
```

Kreiranje novog okvira prijava:

```
public login() { //metoda initComponents();}
```

Metoda za zatvaranje prozora nakon otvaranja novog prozora:

```
public void close() {WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);
Toolkit.getDefaultToolkit().getSystemEventQueue().
postEvent(winClosingEvent);}
```

While petlja za provjeru unesenih podataka u polja za prijavu, u prevodu: sve dok se uneseni podaci u poljima podudaraju sa unesenim podacima iz baze podataka, zatvori trenutni prozor i otvori učenik početna (Home):

```
while(rs.next()) {String uname=rs.getString("username");String password=rs.getString("password"); if ((user.equals(uname)) && (pwd.equals(password))) {close();new Home().setVisible(true);} }
```

Niz naredbi za unos tačnih podataka učenika u mysql tabelu „odgovori“

```
try { Connection con = ( Connection ) DriverManager.getConnection("jdbc:mysql://localhost:3306/proba1","root","123456"); Statement stmt=con.createStatement(); String Selectquery="insert into `proba1`.`odgovori`(`username`,`date`) values ('"+jTextField1.getText()+"',current_timestamp)"; stmt.execute(Selectquery); //izvršava upit iznad }
```

Dugme za otvaranje statistike (profesorst):

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {close(); new profesorst().setVisible(true);}
```

Izvršava sql i bira tabelu lekcija1 i njene podatke:

```
stmt = con.createStatement( ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE ); String sql = "SELECT * FROM proba1.lekcija1"; rs = stmt.executeQuery(sql);
```

Spremanje rezultata (statistika):

```
try { //prikaz tačnih odgovora u poljima String sql11 = " SELECT DISTINCT tacno FROM proba1.lekcija1 where id1=1"; rs = stmt.executeQuery(sql11);rs.next(); String pitanje11 = rs. getString("tacno"); jTextField32.setText(pitanje11);} try{//bodovanje odgovora na pitanje if ( jTextField1. get Text(). equals (jTextField32. getText ())) {jTextField21.setText("1");} else {jTextField21.setText("0");}}
```

Pohranjuje podatke o učeniku i njegovim bodovima u tabelu, kao i vrijeme pohranjivanja:

```
private void jButton1ActionPerformed(java.awt.event.  
ActionEvent evt) {  
try{  
String host = "jdbc:mysql://localhost:3306/proba1";  
String uName = "root";  
String uPass = "123456";  
con = DriverManager.getConnection(host, uName, uPass);  
//izvrsava sql i bira tabelu lekcija1 i njene podatke  
stmt = con.createStatement( ResultSet.TYPE_SCROLL_  
INSENSITIVE, ResultSet.CONCUR_UPDATABLE );  
String sql21 = "SELECT * FROM proba1.statistika";  
rs = stmt.executeQuery(sql21);  
rs.next();  
stmt.executeUpdate( "Insert into `proba1`.`statistika` (`ucenik`,  
`bodovi`,`datum`) values ('"+jLabel2.getText( )+"', '"+jLabel1.  
getText( )+"',current_timestamp)");  
JOptionPane.showMessageDialog(null, "Unos podataka je  
uspjesan!");}}
```

Kratki prikaz dijelova koda oslikava svu složenost u razvoju ovog informacionog sistema i vidi se da je programerski dio posla kompleksan u realizaciji realnih rješenja. Java programski jezik je jako specifičan i fleksibilan za razvoj različitih aplikacija koje rješavaju realne probleme i u kome se mogu razviti kompleksni informacioni sistemi.

Informacioni sistem „AHA!“ može da se nadograđiva u skladu sa potrebama korisnika ovog IS-a. Prvenstveno sljedeća verzija ovog IS-a treba da sadrži modul godišnjih planova, mjesečnih planova i statistike koja bi pratila ishode učenja kod samih učenika.

Zaključak

U ovom radu su prikazani koraci razvoja softvera koristeći model vodopada. Odabran je isključivo zbog jednostavnosti. Ono što nam onemogućuje jeste nadogradnja softvera, što je neprikosnoven nedostatak.

Aplikacija teži sveobuhvatnosti profesorovih i učenikovih aktivnosti, što bi uključivalo nadogradnju više lekcija, kvizova i njihovih tipova. Također bi trebalo težiti kreiranju formulara za

kreiranje dokumentacije za koje profesori inače odvajaju mnogo vremena, a to su: kreiranje i ažuriranje godišnjih planova i programa, mjesečnih obrazaca, nastavnih priprema, kontrolnih radova, evaluacije učenikovih znanja, vođenje vannastavnih aktivnosti itd. Uz to bi se trebalo usmjeriti na sveobuhvatnost platformi na koje se aplikacija može instalirati.

Nadamo se da će ovaj rad prvenstveno imati ulogu u sticanju i proširivanju znanja iz razvoja softvera, te da će nadahnuti studente za kreativno stvaralaštvo.

Literatura

1. David, G. G., Martin F., Myles R. (2013). <http://firstmonday.org/ojs/index.php/fm/article/view/4350/3673#>, (Preuzeto iz Firstmonday, 6.maj 2016)
2. Khattab, M. I. (2015). Employee management system. Nangarhar: Nangarhar University.
3. Michael, T.E., Henry, R. (2013). A Faculty Observation Model for Online Instructors: Observing Faculty Members in the Online Classroom. Preuzeto iz westga: http://www.westga.edu/~distance/ojdla/summer162/eskey_roehrlich162.htm
4. Tomašević, V. (2012). Razvoj aplikativnog softvera. Beograd: Univerzitet Singidunum
5. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3484958/> (Preuzeto iz ncbi: 10.12.2015)

AHA! – APPLICATION FOR LEARNING

Abstract

In the work, it will be analyzed how to improve students' motivation by the means of integrated teaching and appliance of software solutions. The work will also show the phases of design and application development in Java programming environment designed for students and professors, with the aim of learning various content and testing students. During the development of the application, a methodology used in the work is the one that provides a safe path to a solution and facilitates the writing of the code. The application is easy to use and does not require advanced knowledge of computer work, which will create less resistance for students and professors to accept this application.

Keywords: Java, modeling, motivation, quiz, lessons, source code, UML diagrams