Razvoj web okruženja za ocjenjivanje i usporedbu grafičkih korisničkih sučelja

Mandić, Laura

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:021255

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-12**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

RAZVOJ WEB OKRUŽENJA ZA OCJENJIVANJE I USPOREDBU GRAFIČKIH KORISNIČKIH SUČELJA

Diplomski rad

Laura Mandić

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 18.02.2024.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Laura Mandić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1224R, 05.10.2021.
OIB studenta:	09023698257
Mentor:	doc. dr. sc. Krešimir Romić
Sumentor:	doc. dr. sc. Hrvoje Leventić
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Bruno Zorić
Član Povjerenstva 1:	doc. dr. sc. Krešimir Romić
Član Povjerenstva 2:	dr. sc. Ivana Hartmann Tolić
Naslov diplomskog rada:	Razvoj web okruženja za ocjenjivanje i usporedbu grafičkih korisničkih sučelja
Znanstvena grana diplomskog rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Izraditi web okruženje u kojem će ispitanici moći ocijeniti ponuđeno grafičko korisničko sučelje ili pojedinačne elemente istog. Uz ocjenjivanje pojedinačnog korisničkog sučelja omogućiti i usporedbu dva ponuđena korisnička sučelja te zapis rezultata na prikladan način. Omogućiti analizu prikupljenih podataka i njihovo rangiranje. Na primjeru desetak korisničkih sučelja iz baze podataka pokazati rad razvijenog okruženja. Tema rezervirana za: Laura Mandić
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	18.02.2024.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 01.03.2024.

Ime i prezime studenta:	Laura Mandić
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1224R, 05.10.2021.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom**: Razvoj web okruženja za ocjenjivanje i usporedbu grafičkih korisničkih sučelja**

izrađen pod vodstvom mentora doc. dr. sc. Krešimir Romić

i sumentora doc. dr. sc. Hrvoje Leventić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak diplomskog rada 1

2. PREGLED PODRUČJA TEME 3

2.1. Rangiranje igrača 3

2.2. Juxtapeer 3

2.3. Borda algoritam 3

2.4. Algoritam za aproksimaciju ranga 3

2.5. Model procjene korisničkog sučelja i iskustva 4

3. GRAFIČKA KORISNIČKA SUČELJA 5

3.1. Korisničko iskustvo 5

3.1.1. Važnost 5

3.1.2. Vremenski rasponi 6

3.2. Ocjenjivanje grafičkog korisničkog sučelja 7

3.2.1. Evaluacija stručnjaka 8

3.2.2. Testiranje upotrebljivosti 8

3.2.3. Ocjenjivanje vršnjaka 9

3.2.4. Ostale metode 10

4. OKRUŽENJE ZA OCJENJIVANJE I USPOREDBU GRAFIČKIH KORISNIČKIH SUČELJA 11

4.1. Korištene tehnologije 11

4.1.1. .NET 11

4.1.2. C# 11

4.1.3. Angular 12

4.1.4. Azure SQL Database 12

4.2. Arhitektura sustava 12

4.2.1. Slojevi programskog rješenja 12

4.2.2. Model baze podataka 13

4.3. Implementacija rješenja 14

4.3.1. Postavljanje projekta 15

4.3.2. Sloj baze podataka 15

4.3.3. Sloj pristupa podacima 16

4.3.4. Sloj entiteta/modela 16

4.3.5. Sloj poslovne logike ... 17
4.3.6. Prezentacijski sloj ... 18
4.4. Pokaz funkcionalnosti ... 20
4.4.1. Ovjera korisnika ... 20
4.4.2. Grupe dizajna ... 21
4.4.3. Operacije nad dizajnom ... 22
4.4.4. Ocjenjivanje dizajna ... 23
4.4.5. Postavljanje setova usporedbi ... 24
4.4.6. Uspoređivanje dizajna ... 26
4.4.7. Rangiranje dizajna ... 27
4.4.8. Provjera statusa evaluacije ... 28
4.4.9. Izvoz rezultata ... 29
5. EKSPERIMENT ... 30
5.1. Tijek eksperimenta ... 30
5.2. Usporedba ocjena ispitanika i nastavnika ... 30
5.3. Usporedba rangova ispitanika i nastavnika ... 33
5.4. Usporedba ocjena i rangova ... 35
5.5. Usporedba rangova potpune i nepotpune usporedbe ... 35
5.6. Utjecaj smanjenja lambda replikacije ... 36
5.6.1. Smanjenje lambda replikacije na 2 ... 37
5.6.2. Smanjenje lambda replikacije na 1 ... 37
6. ZAKLJUČAK ... 39
LITERATURA ... 40
POPIS KRATICA ... 43
POPIS SLIKA ... 44
SAŽETAK ... 45
ABSTRACT ... 46
ŽIVOTOPIS ... 47
PRILOZI ... 48

1. UVOD

Zahvaljujući konstantnom razvoju tehnologije, sve je šira uporaba digitalnih proizvoda. Tako su danas lako dostupne elektroničke knjige, online tečajevi, nezamjenjivi tokeni (engl. *non-fungible token, NFT*), kriptovalute te razne vrste programske podrške. Mnoge radnje nekada obavljane isključivo uživo danas se premještaju na Internet, poput maloprodaje, kupoprodaje umjetnina, posudbe filmova, slušanja koncerata, kupoprodaje dionica, posjeta liječniku, bankarstva, fitnessa itd. Sve navedene usluge obavljaju se korištenjem određene programske podrške. Kako je ta programska podrška direktni predstavnik tvrtke prema klijentu, tvrtkama je iznimno važno kakve je ona kvalitete zato što uporabom programske podrške korisnik odlučuje hoće li nastaviti koristiti taj digitalni proizvod ili će odabrati drugi [1].

Dizajn korisničkog iskustva i korisničkog sučelja direktno utječe na kvalitetu digitalnog proizvoda. Dizajn korisničkog sučelja utječe na jednostavnost uporabe, čini proizvod intuitivnim i efikasnim te potiče korisnika da se nastavi služiti proizvodom. Usko vezan uz dizajn korisničkog sučelja je dizajn korisničkog iskustva. Kvalitetan dizajn korisničkog iskustva povećava relevantnost i korisnost proizvoda te osigurava da je u svakoj fazi uporabe korisniku omogućen lak pristup sadržaju i funkcionalnostima [2].

Rad je podijeljen na šest poglavlja. Uvodno poglavlje objašnjava zadatak diplomskog rada. Drugo poglavlje obuhvaća pregled područja sačinjen od radova na temu rangiranja i ocjenjivanja korisničkog sučelja te postojećih rješenja. Treće poglavlje pruža teorijski opis grafičkih korisničkih sučelja, njegovih karakteristika te načina ocjenjivanja. Četvrto poglavlje opisuje kreirano okruženje, tehnologije korištene pri njegovom razvoju te samu implementaciju rješenja. Peto poglavlje bavi se provođenjem eksperimenta koji testira uspješnost sustava ocjenjivanja grafičkih korisničkih sučelja i prikladnost ocjenjivanja vršnjaka u dizajnu grafičkih korisničkih sučelja. Posljednje poglavlje opisuje sveukupni rezultat te potencijalne nadogradnje.

1.1. Zadatak diplomskog rada

Zadatak diplomskog rada je izraditi web sučelje koje će omogućiti međusobno ocjenjivanje dizajna grafičkih korisničkih sučelja među vršnjacima – studentima. Aplikacija treba omogućiti postavljanje dizajna na poslužitelj, prikaz skupine dizajna pojedinom studentu preko jedinstvene oznake te prikupljanje rezultata o ocjeni pojedinog dizajna te usporedbi po parovima dizajna. Rezultati usporedbe po parovima kreiraju ukupnu rang listu. Aplikaciji se može pristupiti kroz tri uloge – student, stručnjak ili administrator. Student može dodavati vlastiti dizajn i ocjenjivati i

uspoređivati ostale dizajne. Stručnjak može samo ocjenjivati i uspoređivati dizajne. Administrator ima uvid u ocjene i rang liste svih skupina te može izmijeniti ili dodati podatke čuvane u bazi podataka. Za pristup aplikaciji potrebno je registrirati se u sustav.

2. PREGLED PODRUČJA TEME

2.1. Rangiranje igrača

U [3], Kahng i suradnici sudionicima istraživanja, igračima, predstavljaju parove njihovih kolega konkurenata te ih oni neizravno rangiraju kroz pojedinačne usporedbe parova. U radu se referiraju na mogućnost strateškog ponašanja kojim bi igrač u slučaju predaje gotovog ranga igrača mogao poboljšati svoju poziciju, npr. tako da bolje igrače namjerno rangiraju nisko. U radu se koncentriraju na nepristranost i onemogućavanje igrača da utječu na vlastiti rang, što postižu kroz usporedbe parova. Rezultati su pokazali da je mali broj nasumičnih usporedbi dovoljan za preciznu procjenu stvarnih vještina igrača.

2.2. Juxtapeer

Autori rada predstavljaju Juxtapeer sustav za evaluaciju kroz kojega na nekoliko online kolegija i kolegija izvođenih uživo, od studenata zahtijevaju tekstualnu evaluaciju predanih radova i usporedbu parova. Istražuju doprinosi li usporedno vrednovanje kolega metodi ocjenjivanja vršnjaka. Studenti koji su provodili evaluaciju usporedbom generalno su se više posvetili tekstualnoj evaluaciji i na koncu kvalitetnije proveli vrednovanje. Studija je također pokazala da je rang dobiven kroz usporedbu parova visoko koreliran s rangom eksperata. Komparativni način vrednovanja pokazao se iznimno dobrim u slučaju ocjenjivanja vizualnih radova [4].

2.3. Borda algoritam

U svojemu radu, autori uspoređuju Borda algoritam brojanja pobjeda u usporedbama parova s drugim poznatim modelima izvodeći simulacije na sintetičkim podacima te stvarni eksperiment nad danim setovima podataka. Ispitanici su metodom usporedbe parova morali izabrati sliku s većim brojem stanica čime bi rangirali slike po količini stanica. Nad rezultatima usporedbi izveli su algoritme te došli do zaključka da Borda algoritam izaziva manju Hammingovu pogrešku, tj. manji broj krivo klasificiranih slika. Također, Borda algoritam zahtijeva znatno manje vrijeme izvođenja te se pokazao jednostavnim, robusnim te u konačnici optimalnim [5].

2.4. Algoritam za aproksimaciju ranga

Heckel i suradnici predstavili su aktivni algoritam koji optimizira broj usporedbi parova time što ima zadan kriterij zaustavljanja te prikazuje parove za usporedbu sve dok se kriterij ne ispuni. Također, algoritam tijekom izvođenja odlučuje koji par odabrati idući na temelju intervala

pouzdanosti dobivenih iz ranije donesenih odluka korisnika. Dokazali su da njihov algoritam uspješno aproksimira rang predmeta, a uz to značajno smanjuje broj usporedbi [6].

2.5. Model procjene korisničkog sučelja i iskustva

Autor uvodi novi hibridni programski okvir koji kombinira set algoritama pogonjenih podacima (engl. *data-driven*) za evaluaciju web korisničkog sučelja i iskustva bilo koje stranice. Fokus je na tri ključna indikatora – performanse, dostupnost i korisnost. Performanse i dostupnost automatski su testirane, a korisnost je ocijenjena kroz korisničko ispitivanje. Autor predlaže linearnu jednadžbu kojom se dobiva standardizirani rezultat koji predstavlja kvalitetu korisničkog sučelja i iskustva [7].

3. GRAFIČKA KORISNIČKA SUČELJA

Korisničko sučelje poveznica je između korisnika i sustava kojim oni ostvaruju interakciju potrebnu za upravljanje sustavom, unos/ispis podataka i korištenje sadržaja sustava. Prva verzija korisničkog sučelja bilo je znakovno korisničko sučelje (engl. *Character User Interface, CUI*) koje je obuhvaćalo isključivo tekstualne komande unesene putem komandne linije. Ono je prethodilo grafičkom korisničkom sučelju (engl. *Graphical User Interface, GUI*), koje je nadopunilo znakovna korisnička sučelja raznim grafičkim elementima, poput ikona, prozora, gumba, pokazivača, izbornika, slika i sličnog [8]. Kvalitetno dizajnirano grafičko korisničko sučelje povećava korisnost proizvoda. Zadatak dizajniranja učinkovitog korisničkog sučelja kompleksan je zadatak iz područja interakcije čovjeka i računala (engl. *human-computer interaction, HCI*). Glavni cilj je olakšati uporabu i povećati prilagodljivost sustava kroz dizajn grafičkog korisničkog sučelja [9].

3.1. Korisničko iskustvo

Korisničko iskustvo (engl. *user experience, UX*) odnosi se na cjelokupno iskustvo vezano uz percepciju i reakciju korisnika, njegove osjećaje i misli tijekom korištenja sustava, odnosno aplikacije. Korisničko iskustvo koncept je vezan uz interakciju čovjeka i računala (*HCI*) i primjenjuje se, osim u razvoju softvera i hardvera, i u procesima, proizvodima, uslugama u društvu [8]. Osim toga, za stvaranje pozitivnog korisničkog iskustva potrebno je znanje iz područja dizajna i razvoja proizvoda te psihologije. Obuhvaća ne samo uporabljivost i funkcionalnost proizvoda, nego i efektivnost i efikasnost. Dva su cilja korisničkog iskustva: optimizirati ljudski rad i zadovoljstvo korisnika dostizanjem pragmatičnih i hedonističkih ciljeva. Metode kojima bi se to postiglo ovise o ciljevima proizvoda. Razlikuju se tri kategorije metoda: metode za evaluaciju i dizajn u potrebe hedonističkih ciljeva, evaluaciju i dizajn radi postizanja pragmatičnih ciljeva te podržati dizajn korisničkog iskustva uključivanjem postavljanja zahtjeva i razumijevanjem konteksta korištenja [10].

3.1.1. Važnost

Kada korisnici razviju pozitivne emocije korištenjem proizvoda, izglednije je da će mu se vratiti. Ako ga uključe u svoju rutinu, manje je izgledno da će prijeći na neki drugi proizvod. Važno je da se korisnik osjeća zadovoljno, samopouzdano i cijenjeno kada koristi proizvod jer time raste vjernost i angažiranost [11].

Pristupačnost proizvoda, koja se odnosi na mogućnost korištenja proizvoda unatoč invaliditetima, važna je ne samo pojedinačnom korisniku, nego i iz perspektive zakona i etičkih vrijednosti. Ona obuhvaća korištenje jednostavnog i jasnog jezika, pružanje alternative slikama i konzistentnu navigaciju aplikacijom.

Reputacija aplikacije ili tvrtke utječe na vjernost proizvodu, a gradi se kroz pozitivno oglašavanje međusobnom komunikacijom korisnika. Ako se proizvod jednostavno koristi, lako navigira, i kreira dobro iskustvo, korisnici će sami početi stvarati pozitivnu reputaciju među sobom.

Dobro korisničko iskustvo povećava angažman korisnika, odnosno, vrijeme provedeno koristeći aplikaciju te učestalost vraćanja aplikaciji. Povećanjem angažmana, povećava se i broj klikova te posjećenost, a potencijalno i prihodi.

Angažman korisnika utječe i na optimizaciju pretraživača (engl. *search engine optimization, SEO*), čime će web stranica bolje kotirati u rezultatima pretraživača te privući nove korisnike [12].

Stvaranjem dobrog korisničkog iskustva, tvrtka ostaje kompetitivna. Kako bi se to postiglo, potrebno je omogućiti besprijekorno iskustvo korisnicima. Na taj način korisnici će se osjećati cijenjenima i shvaćenima što vodi do povećanog angažmana i vjernosti proizvodu.

Stvaranjem intuitivne aplikacije smanjuje se i potreba za tehničkom podrškom i održavanjem. Manja je vjerojatnost da će korisnici napraviti pogrešku ako je aplikacija jednostavna za korištenje. Na taj način oslobađaju se resursi koji mogu biti usmjereni poboljšanju aplikacije.

Dobro dizajnirana aplikacija smanjuje potrebu za korisničkom podrškom jer će korisnicima biti lakše pronaći ono što traže te će se sveukupno zadovoljstvo povećati. Visoka potreba za podrškom ne samo da utječe na zadovoljstvo korisnika nego i na potrebu tvrtke za povećanim ljudskim kadrom bez povezanog povećanja prihoda [13].

3.1.2. Vremenski rasponi

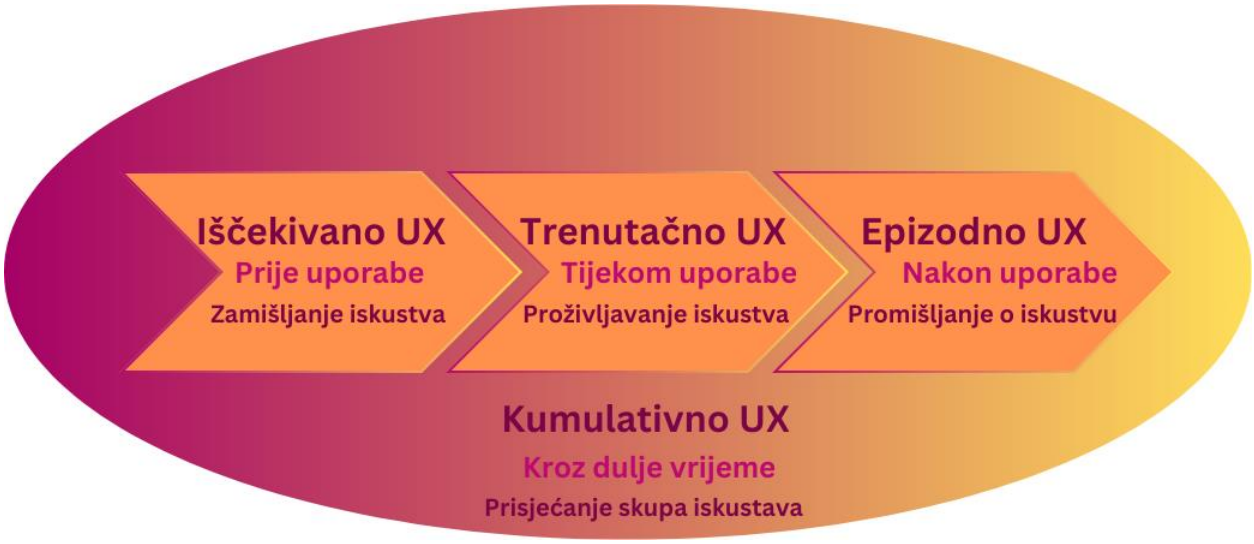
Korisničko iskustvo skuplja se ne samo tijekom uporabe proizvoda nego i prije i nakon uporabe. Također, moguće je promatrati sveukupno korisničko iskustvo kroz cijelo razdoblje korištenja platforme.

Vremenska linija na slici 3.1. počinje iščekivanim korisničkim iskustvom – ono prethodi korištenju proizvoda i formira se kroz slične proizvode, marku proizvoda, reklamiranje i tuđa mišljenja. Prema provedenim istraživanjima, iščekivano iskustvo utječe na buduća korisnička iskustva zbog prilagodbe očekivanog iskustva onom stvarnom [14].

Slijedi trenutačno korisničko iskustvo, ono koje nastaje tijekom interakcije s konačnim proizvodom. Potrebno je promotriti i evaluirati promjene u emotivnom stanju korisnika tijekom uporabe proizvoda.

Nakon korištenja javlja se takozvano epizodno korisničko iskustvo i odnosi se na promišljanje o prethodnoj epizodi uporabe proizvoda te usporedbu s očekivanjima [15].

Korisničko iskustvo koje obuhvaća višestruk prolazak kroz prethodno navedene tri faze naziva se kumulativno korisničko iskustvo. Ono je rezultat slijeda epizoda korištenja te uključuje i nekorištenje kao posljedicu ostalih oblika korisničkog iskustva. Trajanje ovog korisničkog iskustva može biti i nekoliko mjeseci ili godina. U industriji ovo je najvažniji oblik iskustva te prednjači nad kratkotrajnim osjećajima koji se javljaju u pojedinim fazama [16].



Slika 3.1. Vremenska linija korisničkog iskustva

3.2. Ocjenjivanje grafičkog korisničkog sučelja

Evaluacija grafičkih korisničkih sučelja ključna je kako dizajn ne bi bio prilagođen samo onima koji ga kreiraju, već je potrebno ispitivanje šireg tržišta. Faktori koji utječu na način evaluacije uključuju fazu dizajna, razrađenost projekta, očekivani broj korisnika, kritičnost sučelja, troškove namijenjene za testiranje dizajna, dostupno vrijeme te ljudski faktor.

Evaluacijom se ne može postići savršenstvo, niti je moguće u potpunosti dovršiti evaluaciju, jednako kao što nije moguće postići kod testiranja softvera. Pojedine situacije neće biti lako replicirati, a neke se neće moći precizno simulirati, što će utjecati na rezultat ocjenjivanja.

3.2.1. Evaluacija stručnjaka

Najjednostavniji način evaluacije je pregled stručnjaka, koji mogu biti suradnici ili korisnici. Može se provoditi neformalno traženjem mišljenja o dizajnu sustava te u vrlo kratkom razdoblju.

Kognitivni prolazak metoda je kojom stručnjaci simuliraju ulogu korisnika prolazeći aplikacijom. Stručnjaci prate definirani tok zadataka i odgovaraju na zadana pitanja. Prolazak najčešće obavljaju nenadzirano te svoje iskustvo javljaju dizajnerima.

Formalna inspekcija upotrebljivosti odnosi se na široki i skupi postupak prolaska stručnjaka kroz cjelokupni dizajn aplikacije gdje svaka strana argumentira prednosti i nedostatke sučelja, a neutralna strana donosi odluku.

Pregled stručnjaka opcija je koja se može provesti više puta u procesu dizajna. Fokus se stavlja na sveobuhvatnost umjesto konkretnih značajki i nasumičnih poboljšanja. Prijedlozi za ispravcima klasificiraju se po važnosti te se manje važni popravci mogu odgoditi za kasnije faze. Stručnjaci trebaju iskušati sučelje u stvarnim uvjetima iz pozicije korisnika. Pogled iz ptičje perspektive također može biti od koristi tako da se čitav komplet ekrana pregledava istovremeno jedan uz drugi [17].

3.2.2. Testiranje upotrebljivosti

Testiranje upotrebljivosti u laboratoriju omogućuje inspekciju u kontroliranim uvjetima s odabranim uzorkom korisnika. Ono omogućava brži razvoj projekata i smanjenje troškova. Postiže se praktičnije istraživanje, koje zamjenjuje isključivo teoretske rasprave. Za laboratorijsko testiranje potreban je kadar koji će uz pomoć dizajnera sučelja napraviti plan testiranja, popis zadataka te broj, tip i izvor ispitanika. Potrebno je bilježiti ponašanje korisnika tijekom uporabe proizvoda radi naknadne analize.

Testiranje upotrebljivosti dobro je izvoditi u ranim fazama i često jer se najozbiljniji problemi mogu detektirati uz samo nekolicinu ispitivača. Takav pristup naziva se olakšano testiranje upotrebljivosti i nije upotrebljivo kod velikih sustava [18].

Kompetitivno testiranje upotrebljivosti uspoređuje trenutnu verziju sučelja s prethodnim verzijama ili verzijom kompetitora. Prilikom usporedbe, najbolja je opcija usporediti sučelje s drugim sučeljima iste tematike, te voditi računa o izmjeni slijeda prikaza sučelja.

Testiranje univerzalne upotrebljivosti za cilj ima omogućiti korisnicima pristup informacijama i komunikacijskim uslugama. Tri su primarna izazova s kojima se suočava postizanje univerzalne

upotrebljivosti: tehnološka raznolikost, korisnička raznolikost i rupe u znanju. Tehnološka raznolikost obuhvaća različite platforme, hardver i mreže. Potrebno je izvršiti testiranje uz visoku raznolikost kako bi se simulirali stvarni uvjeti [19].

Testiranje upotrebljivosti ograničeno je na početak korištenja aplikacije te korisnike koji se po prvi puta služe proizvodom. Kod nekih proizvoda teško je simulirati stvarne uvjete, poput onih kada su korisnici pod velikim pritiskom. Unatoč nedostacima, studije su pokazale da testiranje upotrebljivosti utječe na uštedu sredstava i povrat uloženog [20].

3.2.3. Ocjenjivanje vršnjaka

Ocjenjivanje vršnjaka odnosi se na metodu gdje dizajneri koji su međusobno ravnopravni, često učenici ili studenti, analiziraju radove jedni drugih te pružaju povratne informacije o njima [21]. Ocjenjivanje vršnjaka najveću je vrijednost pokazalo kod masivnih otvorenih online kolegija (engl. *massive open online course, MOOC*) jer se radi o skalabilnom i održivom načinu evaluacije. Empirijski rezultati istraživanja pokazali su da skupne ocjene dobivene od kolega pretežito streme ocjenama nastavnika, a da su studenti dobro prihvatili takav način ocjenjivanja te su ga smatrali korisnim i poštenim [22].

Ocjenjivanje vršnjaka može se podijeliti na ordinalno i kardinalno. Kardinalno ocjenjivanje pridružuje određenu veličinu, najčešće ocjenu, pojedinom dizajnu, dok ordinalno dodjeljuje poziciju u poretku svakome dizajnu. Ordinalno ocjenjivanje pokazalo se prikladnijim načinom ocjenjivanja jer onemogućuje dodjeljivanje jednake ocjene svim radovima [23].

Parno uspoređivanje

Parno uspoređivanje (engl. *pairwise comparison, PWC*) metoda je ordinalnog ocjenjivanja. Ocjenjivači promatraju dva po dva dizajna te donose odluku koji je od njih bolji. Kao rezultat dobiva se takozvana matrica parne usporedbe (tablica 3.1.) koja prikazuje sve promatrane dizajne i dimenzija je n×n, gdje n predstavlja broj promatranih dizajna. Pojedini redak iskazuje broj pobjeda dizajna retka nad dizajnom stupca. Iz ukupnog broja ili postotka pobjeda u usporedbama izvodi se rang lista dizajna (tablica 3.2.).

Tablica 3.1. Matrica parne usporedbe

	A	**B**	**C**
A	-	0	0
B	1	-	0
C	1	1	-

Tablica 3.2. Rang lista izvedena iz matrice parne usporedbe

1. Dizajn C
2. Dizajn B
3. Dizajn A

3.2.4. Ostale metode

Provođenje ispitivanja alat je koji omogućava prikupljanje povratne informacije i evaluaciju grafičkog korisničkog sučelja putem upitnika, intervjua, alfa/beta testiranja i viševarijantnog testiranja. Pri odabiru pitanja za istraživanje, bilo bi dobro izabrati pitanja koja su pregledali stručnjaci, koja su pilot testirana na malom uzorku i temelje se na dobro ispitanim postojećim pitanjima [24].

Testiranje prihvatljivosti izvršava se na cjelokupnom sustavu i provjerava je li sustav prihvatljiv korisniku. Postavljaju se objektivni i mjerljivi ciljevi na performanse korisnika, poput vremena potrebnog da korisnik nauči koristiti funkciju proizvoda, učestalosti pogrešaka, brzine izvođenja zadataka [25] itd.

Evaluacija tijekom aktivnog korištenja još je jedna od opcija, a moguće ju je provoditi putem intervjua s pojedincima, raspravama u fokusnim skupinama, kontinuiranim praćenjem (engl. *logging*) i slično.

Korisnička podrška također može prikupljati podatke o problemima s kojima se korisnici suočavaju paralelno s pružanjem pomoći tim istim korisnicima. Radi se o nejeftinom načinu evaluacije, ali osjetno utječe na zadovoljstvo korisnika [17].

4. OKRUŽENJE ZA OCJENJIVANJE I USPOREDBU GRAFIČKIH KORISNIČKIH SUČELJA

Poglavlje opisuje proces razvoja web okruženja za evaluaciju grafičkih korisničkih sučelja, počevši od primijenjenih tehnologija. Zatim prikazuje arhitekturu sustava po slojevima i model baze podataka. Nadalje se objašnjava postavljanje projekata po slojevima te navodi uloga pojedinog sloja i njegove glavne karakteristike.

4.1. Korištene tehnologije

4.1.1. .NET

.NET je Microsoftova razvojna platforma otvorenog koda namijenjena za stvaranje aplikacija raznih vrsta. Temelji se na Common Language Runtime (*CLR*) izvršnom okruženju koje vodi brigu o prikupljanju smeća (engl. *garbage collection*), sigurnom pristupu memoriji te tipskoj sigurnosti (engl. *type-safety*). CLR je prenosiv neovisno o platformi, operativnom sustavu i arhitekturi. Za pokretanje .NET koda stoga nije potrebno ponovno kompajliranje koda, već samo instalacija drugog izvršnog okruženja. CLR podržava programske jezike C#, F# i Visual Basic. Za dodavanje kompajliranih binarnih datoteka u projekte, koristi se NuGet upravljač paketima. .NET, u početku zvan .NET Core, nasljednik je razvojne platforme .NET Framework, a razvijen je kako bi se postigla modularnost, fleksibilnost i prenosivost u doba računarstva u oblaku [26].

4.1.2. C#

Za razvoj poslužiteljskog dijela aplikacije korišten je programski jezik C#, jedan od najpopularnijih objektno-orijentiranih jezika. C# strogo je tipizirani (engl. *strongly typed*) jezik s unificiranim sustavom tipova. Svi tipovi, uključujući one primitivne, nasljeđuju iz korijenskog tipa „*object*“. Podržava korisnički definirane referentne tipove (engl. *reference type*) i vrijednosne tipove (engl. *value type*). Zahvaljujući svojim karakteristikama prikupljanja otpada, ništavnih tipova (engl. *nullable type*) i rukovanju pogreškama, pojednostavljuje stvaranje robusnih aplikacija. U jezik su integrirani LINQ (*Language-Integraded Query*) upiti koji omogućavaju korištenje uniformne sintakse za sve upite, neovisno o tehnologiji i bazi podataka. Omogućavaju filtriranje, grupiranje i sortiranje uz vrlo malo koda [27], a za razvoj aplikacije u ovome radu korištena je sintaksa temeljena na metodama, primjer i usporedba s regularnom sintaksom prikazane su slikom 4.1.

```
var standardQuery = from product in products
             where product.Id < 3
            orderby product.Name
            select product;

var methodQuery = products.
    Where(product => product.Id < 3).
    OrderBy(product => product.Name);
```

Slika 4.1. Usporedba sintakse LINQ

4.1.3. Angular

Angular je razvojni programski okvir (engl. *framework*), temeljen na TypeScriptu i stvoren u Google-u kao zamjena za AngularJS. Otvorenog je koda, i služi za stvaranje jednostraničnih aplikacija (engl. *single-page application, SPA*) što omogućava glađe tranzicije sadržaja. Gradivni blokovi su komponente, a svaka ima HTML predložak, TypeScript klasu, CSS selektor te najčešće i CSS stilove. Jedan od osnovnih koncepata je ubrizgavanje ovisnosti (engl. *dependency injection*) koji je ugrađen u razvojni okvir i omogućava klasama s Angular dekoratorima da koriste ovisnosti koje im trebaju [28].

4.1.4. Azure SQL Database

Azure SQL Database relacijska je baza podataka u oblaku. Samoupravljiva je (engl. *fully managed*) što znači da sama vodi računa o sigurnosnim kopijama, nadogradnjama i nadzoru. Pokreće se na najnovijoj stabilnoj verziji SQL Server engine-a [29].

4.2. Arhitektura sustava

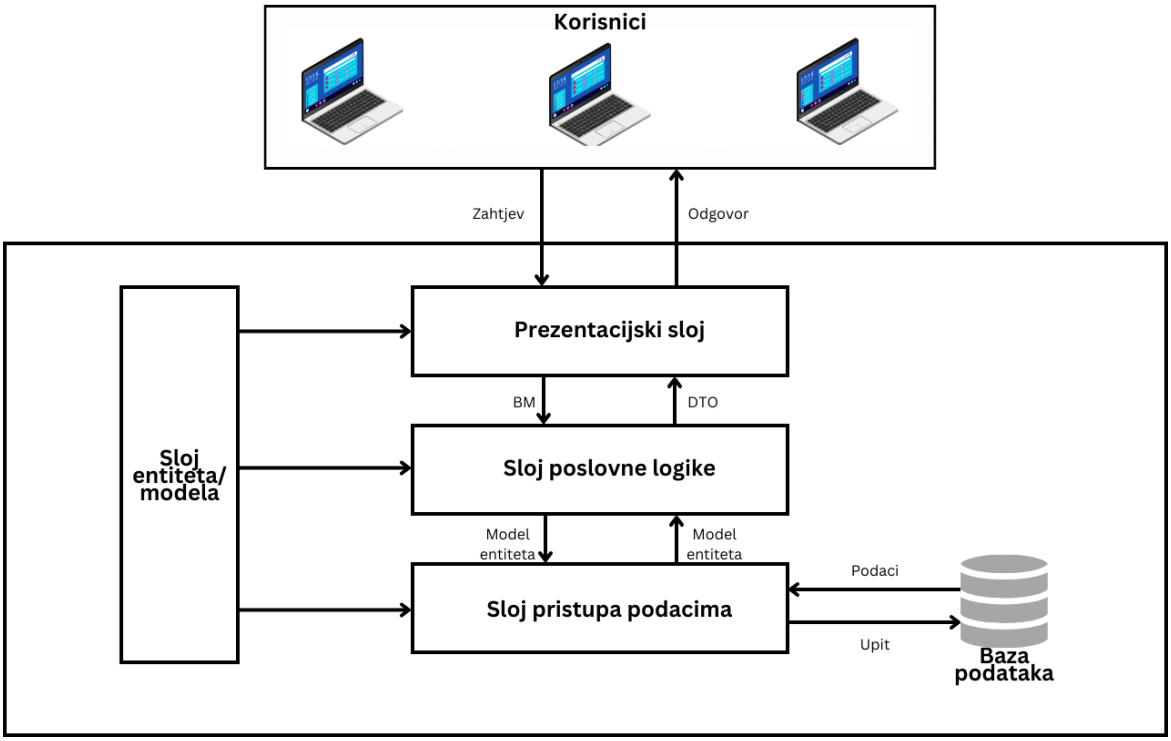
4.2.1. Slojevi programskog rješenja

Sustav je podijeljen u slijedećih 5 slojeva (slika 4.2.):

- Sloj SQL baze podataka – sadrži definiciju baze i svih njenih dijelova
- Sloj pristupa podacima (engl. *Data Access Layer, DAL*) – sadrži definiciju klasa koje predstavljaju entitete baze podataka te njihove odnose
- Sloj poslovne logike – sadrži metode za obradu klijentskih zahtjeva
- Sloj entiteta/modela – sadrži definicije objekata koje izmjenjuju klijent i poslužitelj, tzv. objekte za prijenos podataka (engl. *data transfer object, DTO*) i modele vezivanja (engl. *binding model, BM*)

- Prezentacijski sloj – sadržava klijentsku aplikaciju i API (engl. *application programming interface*) kontroler koji je posrednik između klijenta i poslužitelja.

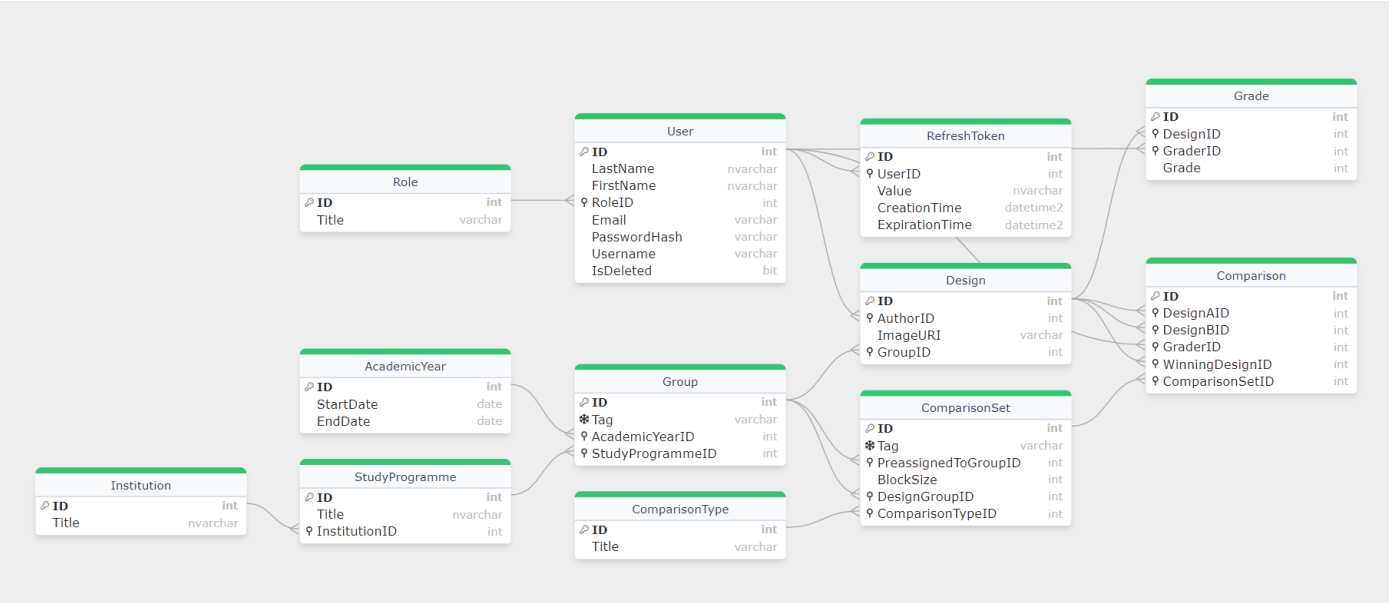
Zahtjeve korisnika predane kroz klijentsku aplikaciju obrađuje API kontroler na prezentacijskom sloju. Zahtjevi su pretvoreni u modele vezivanja koje kontroler predaje sloju poslovne logike. Tu se obavlja obrada i pristup podacima kroz mapiranje modela vezivanja na model entiteta. Koristeći model entiteta šalju se SQL upiti na bazu podataka. Baza podataka odgovara podacima koji se pretvaraju u model entiteta. Zatim sloj poslovne logike obavlja daljnju obradu i šalje odgovor klijentskoj aplikaciji u obliku objekta za prijenos podataka. U konačnici se odgovor korisniku prikazuje na korisničkom sučelju aplikacije.



Slika 4.2. Slojevi programskog sustava

4.2.2. Model baze podataka

Za spremanje podataka i operacije nad njima korištena je Azure mrežna baza podataka. Podaci su modelirani dijagramom entiteta i veza (engl. *entity-relationship diagram, ER diagram*) prikazanim slikom 4.3. Entitet Grade predstavlja podatke o dodijeljenim ocjenama kardinalnim ocjenjivanjem, a entitet Comparison predstavlja rezultat ordinalnog ocjenjivanja, odnosno informaciju o bolje rangiranom dizajnu u direktnoj usporedbi dva dizajna.



Slika 4.3. Dijagram entiteta i veza

4.3. Implementacija rješenja

Cilj je implementirati web aplikaciju koja će omogućiti registraciju, prijavu i odjavu korisnika, dodavanje dizajna u grupu, dodavanje, pregled, izmjenu i brisanje korisnika, smjerova, akademskih godina i institucija. Potrebno je omogućiti kardinalno ocjenjivanje pridruživanjem ocjene od 1 do 5 dizajnu, kao i ordinalno ocjenjivanje parnom usporedbom dva po dva dizajna. Administratoru se mora omogućiti pregled statusa ocjenjivanja studenata, kao i uvid u rezultate i dizajne prenesene na poslužitelj. Administrator mora moći stvoriti novi set dizajna za usporedbu prilagođen načinu usporedbe koji želi provesti – potpuni ili nepotpuni. Glavne funkcionalnosti prikazane su tablicom 4.1.

Tablica 4.1. Pregled funkcionalnosti

Korisnici	**Dizajn**	**Evaluacija**	**Rezultati**
Registracija	Dodavanje	Ocjenjivanje	Tijek evaluacije
Prijava	Pregled	Parna usporedba	Ukupne ocjene
Odjava	Izmjena	Stvaranje seta usporedbi	Rang lista
Pregled	Brisanje		Izvoz rezultata
Izmjena			
Brisanje			

4.3.1. Postavljanje projekta

Koristeći Visual Studio integrirano razvojno okruženje stvoren je „.NET Core with Angular“ predložak. Programsko rješenje (engl. *solution*) podijeljeno je na 5 slojeva, odnosno projekata prikazanih slikom 4.2.

Prezentacijski sloj ima referencu na sloj poslovne logike i sloj entiteta, a sloj poslovne logike ima referencu na sloj pristupa podacima i sloj entiteta.

4.3.2. Sloj baze podataka

Sloj baze podataka kreira se pomoću SchemaCompare datoteke koja je dio SQL Server Data Tools alata i omogućuje usporedbu dvije definicije baza podataka. Izvor i meta usporedbe mogu biti bilo koja kombinacija spojene baze podataka, projekta baze podataka, snimka (engl. *snapshot*) baze podataka ili .dacpac datoteke. Rezultat usporedbe je set postupaka koji se moraju poduzeti da bi meta postala jednaka izvoru. Nakon usporedbe, moguće je napraviti izmjene na meti direktno iz okruženja ili generirati SQL skriptu za izmjenu na bazi [30]. U ovome slučaju, izvor je spojena Azure baza podataka, a meta projekt baze podataka. Primjer dobivene skripte prikazan je slikom 4.4.

```
CREATE TABLE [dbo].[AcademicYear] (
    [ID]        INT  IDENTITY (1, 1) NOT NULL,
    [StartDate] DATE NOT NULL,
    [EndDate]   DATE NOT NULL,
    PRIMARY KEY CLUSTERED ([ID] ASC)
);
```

Slika 4.4. SQL skripta za stvaranje tablice

Za spajanje na Azure bazu podataka i spremište potrebno je dodati stringove veze (engl. *connection string*) u appsettings.json datoteci na prezentacijskom sloju. Kako je za pristup resursima potrebna zaporka, njeno izlaganje spriječeno je koristeći korisničke tajne (engl. *user secrets*) u fazi razvoja i sefa s ključevima (engl. *key vault*) u fazi produkcije. Prilikom pokretanja aplikacije poziva se metoda koja postavlja ovisnosti, dodaje zaporku u stringove veze i inicijalizira kontekst baze podataka, a dio metode prikazan je na slici 4.5.

```
var builder = new SqlConnectionStringBuilder(
          configuration.GetConnectionString("GuiDB"));
builder.Password = configuration["DbPassword"];

var connection = builder.ConnectionString;
services.AddDbContext<GUIGradingDatabaseContext>(options =>
       {
           options.UseSqlServer(connection,
             sqlServerOptions => sqlServerOptions.CommandTimeout(300));
       }, ServiceLifetime.Transient);
```

Slika 4.5. Spajanje na bazu podataka

4.3.3. Sloj pristupa podacima

Za kreiranje sloja pristupa podacima koristi se Visual Studio ekstenzija EF Core Power Tools, odnosno njen Reverse Engineer alat koji kreira DbContext klasu i klase entiteta na temelju projekta baze podataka. Tako postaje moguće koristiti Entity Framework Core i LINQ za rad s bazom podataka. Primjer generirane klase entiteta prikazan je slikom 4.6.

```
public partial class AcademicYear
    {
        public AcademicYear()
        {
            Group = new HashSet<Group>();
        }

        public int Id { get; set; }
        public DateTime StartDate { get; set; }
        public DateTime EndDate { get; set; }

        public virtual ICollection<Group> Group { get; set; }
    }
```

Slika 4.6. Automatski generirana klasa entiteta

4.3.4. Sloj entiteta/modela

Sloj entiteta pruža modele podataka koji se razmjenjuju između klijentske i poslužiteljske aplikacije. On omogućava enkapsulaciju neophodnih podataka čime se izbjegava slanje osjetljivih i suvišnih podataka klijentu te se definira koje podatke poslužitelj očekuje od klijenta. Modeli se dijele na objekte za prijenos podataka i modele vezivanja. Podaci za prijenos podataka koriste se u smjeru komunikacije od poslužitelja prema klijentu, a modeli vezivanja za komunikaciju od klijenta prema poslužitelju. Primjer modela vezivanja za akademsku godinu prikazan je slikom 4.7. Sve te modele potrebno je mapirati na objekte klasa sloja pristupa podacima kako bi ih se

moglo koristiti u operacijama nad bazom podataka. Za mapiranje koristi se NuGet AutoMapper i klasa MappingProfile koja definira mapiranja.

```
public class AcademicYearBM
    {
        [Required]
        public string StartDate { get; set; }
        [Required]
        public string EndDate { get; set; }

    }
```

Slika 4.7. Model vezivanja za akademsku godinu

Kod mapiranja modela entiteta na DTO (slika 4.8.), sva svojstva koja nemaju međusobno identične nazive ili su različitih tipova podataka, potrebno je eksplicitno definirati na koju se vrijednost to svojstvo postavlja.

```
CreateMap<User, UserInfoDTO>()
                .ForMember(d => d.Role, o => o.MapFrom(src => src.Role.Title));
```

Slika 4.8. Mapiranje modela entiteta na DTO

Kod mapiranja modela vezivanja na modele entiteta (slika 4.9.) moraju se definirati sva svojstva koja model entiteta ima, a ukoliko ih nije potrebno definirati, odnosno ako se postavljaju na vrijednost *null*, to je potrebno odrediti pomoću Ignore() metode.

```
CreateMap<AcademicYearBM, AcademicYear>()
                .ForMember(d => d.Id, o => o.Ignore())
                .ForMember(d => d.Group, o => o.Ignore());
```

Slika 4.9. Mapiranje modela vezivanja na model entiteta

4.3.5. Sloj poslovne logike

Sloj poslovne logike temelji se na servisima koji se injektiraju u kontrolere. Servisi izlažu svoje javne metode preko sučelja koja implementiraju, dok su metode koje kontroleri ne pozivaju postavljene privatnima i nisu definirane sučeljima.

```
public AcademicYearDTO GetYearByID(int id)
        {
            var year = context.AcademicYear.SingleOrDefault(p => p.Id == id);
            if (year == null)
            {
                throw new CustomizedException("Invalid year ID.");
            }
            return mapper.Map<AcademicYearDTO>(year);
        }
```

Slika 4.10. Javna metoda servisa AcademicYearService

Ubrizgavanje ovisnosti obavlja se tijekom pokretanja aplikacije i prikazano je slikom 4.11., kada se sve servise dodaje u kolekciju sa *Scoped* životnim vijekom, odnosno koristi se isti objekt tijekom obrade jednog HTTP zahtjeva [31].

```
services.AddScoped<IAcademicYearService, AcademicYearService>();
```

Slika 4.11. Ubrizgavanje ovisnosti

4.3.6. Prezentacijski sloj

Prezentacijski sloj obuhvaća inicijalno dodan .NET + Angular projekt. U njemu se nalaze API kontroleri, međuprogramska oprema (engl. *middleware*), Startup klasa, appsettings.json datoteka i klijentska aplikacija.

API kontroleri

API kontroleri čekaju zahtjeve usmjerene na definirane URL adrese. Kontroler ima korijensku adresu na koju se nastavljaju specificirane adrese za pojedinu metodu kontrolera. Kontroleri

```
[HttpGet("{id}")]
public IActionResult GetAcademicYearById([FromRoute] int id)
        {
            var result = yearService.GetYearByID(id);
            return Ok(result);
        }
```

Slika 4.12. POST metoda za dodavanje akademske godine

provjeravaju autorizaciju. Za pojedine metode autorizacija nije potrebna, za druge je potrebno samo da je korisnik prijavljen, a za ostale je važna i uloga korisnika. Na kontroleru je uz svaku adresu definirana i HTTP metoda zahtjeva koja se mora koristiti.

Autorizacija može biti postavljena za čitav kontroler ili za svaku metodu pojedinačno. Za autorizaciju koristi se prilagođeni Authorize atribut. Ako je potrebno provjeriti ulogu korisnika, dopuštene uloge se predaju kao parametri atributa.

Podaci koje klijentska aplikacija šalje mogu se prenijeti u nekoliko različitih oblika, a prilikom definiranja parametara potrebno je atributom navesti odakle oni potječu, odnosno u kojem se obliku šalju. Atributi koji se mogu koristiti su:

- [FromQuery] – dohvaća vrijednosti iz stringa upita
- [FromRoute] – dohvaća vrijednosti iz adrese
- [FromForm] – dohvaća vrijednosti iz forme
- [FromBody] – dohvaća vrijednosti iz tijela zahtjeva

- [FromHeader] – dohvaća vrijednosti iz HTTP zaglavlja [32].

Međuprogramska oprema

Međuprogramska oprema se uključuje u obradu HTTP zahtjeva po primitku zahtjeva, a prije glavne obrade, ili nakon obrade, a prije slanja odgovora. U ovom slučaju, predobrada se odnosi na autorizaciju, odnosno korištenje Authorize atributa te čitanje JWT (*JSON Web Token*) tokena koji se nalazi u zaglavlju zahtjeva i validaciju podataka iz tokena uz pomoć skrivenog ključa. Postobrada uključuje postavljanje HTTP statusnog koda kod pojave greške.

Startup klasa

Startup klasa karakteristična je za .NET Core aplikacije i ona konfigurira servise i cjevovod (engl. *pipeline*) obrade zahtjeva. Startup klasa specificira se kada se izgrađuje *host* aplikacije. *Host* pruža servise dostupne konstruktoru klase Startup, a klasa dodaje ostale servise kroz ConfigureServices metodu.

Datoteka appsettings.json

Datoteka appsettings.json izvor je konfiguracije koji sadržava ključ-vrijednost parove koji se učitavaju u konfiguraciju aplikacije. Moguće je kreirati datoteke za pojedini tip okruženja, poput razvoja ili produkcije. U tome slučaju vrijednosti spremljene u datotekama za pojedino okruženje pregaze vrijednosti s istim ključem u appsettings.json datoteci. Sadržava podatke o bilježenju problema, stringove veze, nazive sefa ključeva itd.

Klijentska aplikacija

Klijentska aplikacija pisana u Angularu definira se unutar datoteke proizvoljnog imena koje se navodi u projektnoj datoteci pod SpaRoot oznakom. Pri stvaranju Angular aplikacije, generirane su datoteke radnog prostora (engl. *workspace*):

- angular.json – sadržava konfiguracijske opcije za *build*, *serve* i *test* alate koje koristi Angular CLI (engl. *command line interface*)
- package.json – sadržava konfiguracije paketa ovisnosti
- package-lock.json – sadržava informacije o verzijama svih paketa instaliranih u node_modules

- node_modules/ – pruža npm (engl. *node package manager*) pakete čitavom radnom prostoru
- src/ - korijenske datoteke aplikacije
- tsconfig.json – sadrži konfiguraciju za TypeScript
- README.md – uvodna dokumentacija o aplikaciji
- .gitignore – specificira datoteke koje Git ne treba pratiti
- .editorconfig – sadrži konfiguraciju za uređivače koda [33].

Klijentska aplikacija pokreće se komandom *ng serve* uz proizvoljne opcije poput konfiguracije okruženja, otvaranja stranice u pretraživaču, specifikacije vrata (engl. *port*) itd.

4.4. Pokaz funkcionalnosti

4.4.1. Ovjera korisnika

Za ovjeru korisnika koriste se tokeni – JSON Web Token (*JWT*) i tokeni za osvježavanje (engl. *refresh token*). JWT se izdaje pri svakoj prijavi i registraciji u sustav i validan je 30 minuta od izdavanja. Sastoji se od tri dijela – zaglavlja, podataka i potpisa međusobno razdvojenih točkama i enkodiranih po base64 shemi. Zaglavlje definira algoritam šifriranja potpisa i vrstu tokena, u ovome slučaju koristi se HMAC SHA-256 algoritam. Podaci sadrže enkodirane parove ključ-vrijednost po izboru, ovdje ID korisnika, njegovo korisničko ime i ulogu. Svakom odlaznom HTTP zahtjevu, izuzev onima za prijavu i registraciju, u zaglavlje presretač (engl. *interceptor*) dodaje JWT (slika 4.13.).

Kada poslužitelj primi zahtjev, JWT međuprogramska oprema provjerava validnost tokena pomoću njegovog potpisa kako bi se provjerilo da nitko nije mijenjao vrijednosti u podatkovnom dijelu. Ukoliko token nije validan, poslužitelj šalje HTTP odgovor s 401 *Unauthorized* statusnim kodom.

```
if (isLoggedIn && isApiUrl) {
        request = request.clone({
          setHeaders: { Authorization: `${jwtToken}` },
        });
      }
```

Slika 4.13. Dodavanje autorizacijskog zaglavlja HTTP zahtjevu

Tokeni za osvježavanje duljeg su trajanja nego JWT, ističu 24 sata nakon izdavanja. Služe kako bi korisnik mogao dobiti novi JWT bez ponovne prijave. Nakon prijave i registracije, na klijentskoj aplikaciji pokreće se *timer* koji okida osvježavanje JWT tokena (slika 4.14.). Jednu minutu prije isticanja JWT tokena, u pozadini se šalje zahtjev za izdavanjem novog JWT tokena i prilaže token za osvježavanje.

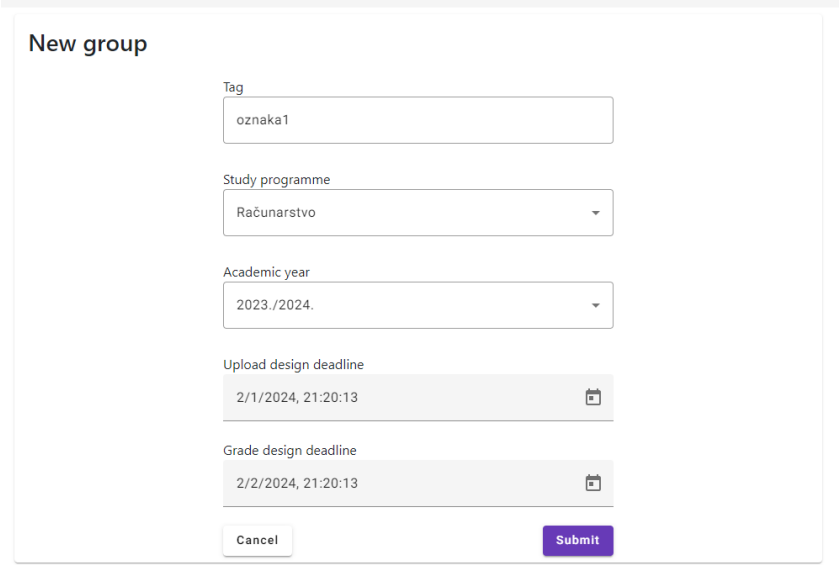
```
public login(model: LoginBM) {
    return this.httpClient
      .post<AccountDto>(`${this.serviceBaseUrl}/login`, model)
      .pipe(
        map((user) => {
          this.tokenService.userSubject.next(user);
          this.startRefreshTokenTimer();
          return user;
        })
      );
  }
```

Slika 4.14. Postupak prijave na klijentskoj aplikaciji

Ako je token za osvježavanje validan, poslužitelj će generirati novi JWT. Svakom se prijavom uz JWT, izdaje i novi token za osvježavanje, čime se postiže rotacija tokena kao dodatna sigurnosna mjera.

4.4.2. Grupe dizajna

Dizajni studenata sortiraju se u grupe s unikatnim oznakama (engl. *tag*) unutar kojih se obavlja ocjenjivanje i usporedba. Administrator odlučuje kako će raspodijeliti studente, odnosno njihove dizajne u grupe. Pri dodavanju grupe, potrebno je definirati njenu oznaku, smjer studija i akademsku godinu. Uz to, definiraju se rok za učitavanje dizajna na poslužitelj i rok za ocjenjivanje i usporedbu. Obavljanje tih zadatka onemogućeno je nakon roka. Administrator može naknadno promijeniti podatke o grupi kao i rokove.

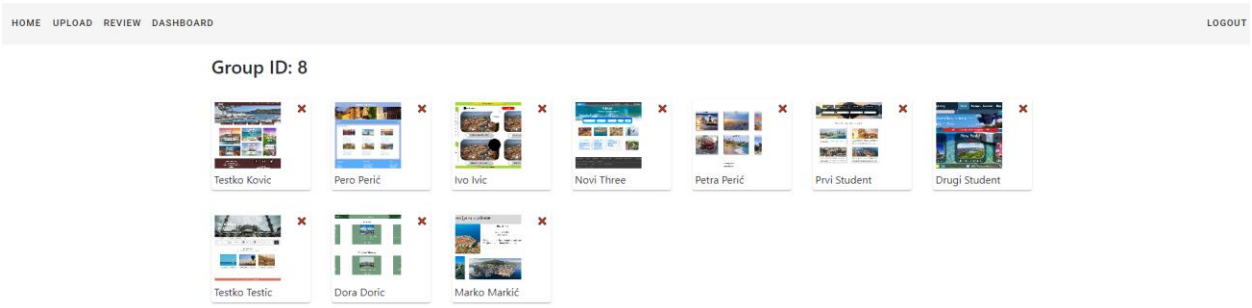


Slika 4.15. Dodavanje grupe

4.4.3. Operacije nad dizajnom

Dizajn u grupu može dodati student ili administrator. Student mora znati oznaku grupe unaprijed kako bi mogao dodati svoj dizajn te ukoliko pokuša dodati još jedan dizajn u istu grupu, stari dizajn bit će prebrisan. Administrator može dodati cijelu grupu dizajna, pri čemu treba specificirati autora svakog dizajna. U slučaju kada administrator dodaje dizajne, više dizajna može imati istog autora. Administrator pri dodavanju dizajna može pretraživati grupe i korisnike.

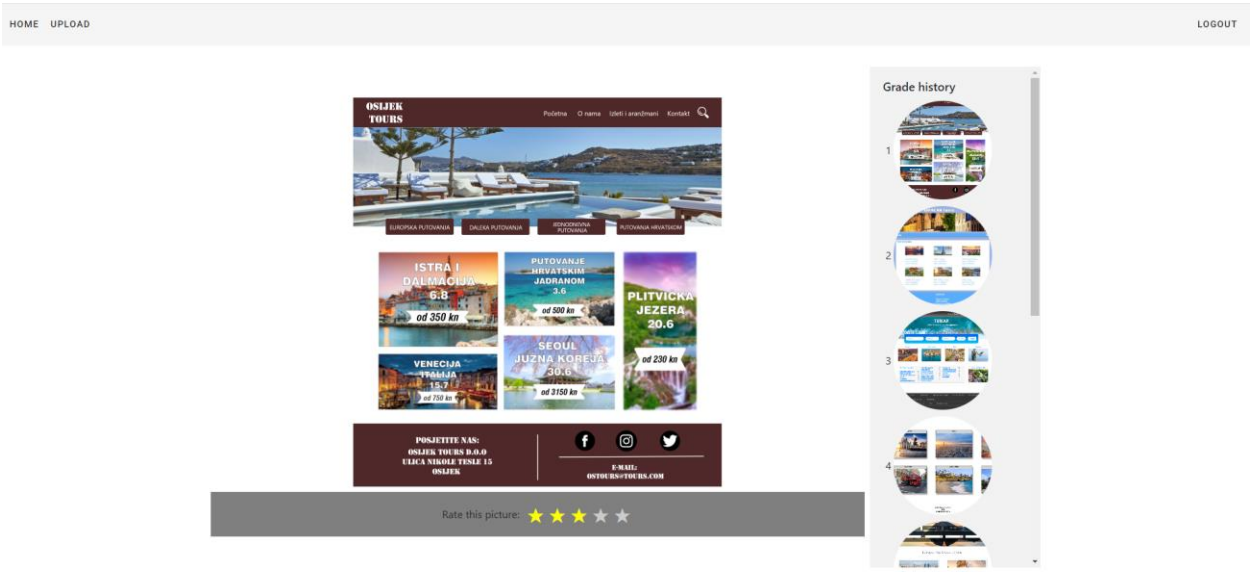
Administrator preko nadzorne ploče može pregledati sve dizajne grupe i potencijalno ih brisati (slika 4.16.). Studenti ne mogu obrisati svoj dizajn, već samo zamijeniti postojeći novim, unutar zadanog roka. Administratoru su sve radnje dopuštene i izvan zadanih rokova.



Slika 4.16. Pregled dizajna grupe

4.4.4. Ocjenjivanje dizajna

Ocjenjivanje dizajna omogućuje kardinalnu evaluaciju čime se ocjenjivači ne obvezuju na postavljanje redoslijeda dizajna nego više dizajna mogu smatrati i ocijeniti jednako dobrima (slika 4.17.). Grupe dizajna može ocijeniti bilo koji korisnik sustava dokle god mu je poznata oznaka grupe dizajna. Ukoliko je korisnik ujedno autor jednog ili više dizajna, njih neće dobiti na ocjenjivanje. Svakom dizajnu dodjeljuje se ocjena od 1 do 5 i moguće ju je promijeniti sve do spremanja u bazu. Prije spremanja ocjena u bazu podataka potrebno je ocijeniti sve dizajne iz grupe. Ukoliko korisnik želi pauzirati i naknadno se vratiti ocjenjivanju, može kasnije nastaviti tamo gdje je stao, dokle god se ne odjavi jer se svaka ocjena sprema u lokalnu pohranu (engl. *local storage*). Podaci tako ostaju spremljeni čak ako se i prozor preglednika ugasi, a korisnik ostane prijavljen. Ukoliko prođe vrijeme isteka JWT tokena, a aplikacija nije pokrenuta, token neće biti osvježen, a korisnik će biti automatski odjavljen idućim pristupom. Ocjenjivanje je ograničeno rokom zadanim na grupi.



Slika 4.17. Ocjenjivanje grupe dizajna

4.4.5. Postavljanje setova usporedbi

Postoje 3 vrste setova usporedbi:

- set potpune usporedbe – ocjenjivač dobiva cijeli skup usporedbi kojim se uspoređuje svaki dizajn sa svim ostalim u pojedinoj grupi
- set nepotpune nedodijeljene usporedbe – ocjenjivač nije dio prethodno definirane grupe i podskup svih usporedbi dizajna mu se dodjeljuje kada pokrene postupak uspoređivanja
- set nepotpune dodijeljene usporedbe – ocjenjivač je dio prethodno definirane grupe i podskup svih usporedbi dizajna mu se dodjeljuje prije pokretanja postupka uspoređivanja.

Set potpune usporedbe automatski se stvara dodavanjem grupe i poprima oznaku grupe. Kod potpune usporedbe uspoređuje se svaki dizajn sa svakim drugim dizajnom u pojedinoj grupi, uz izostavljanje dizajna čiji je autor ocjenjivač. Time je broj usporedbi zadanih ocjenjivaču jednak $\frac{n \cdot (n-1)}{2}$, gdje je n ukupan broj dizajna grupe kojima autor nije ocjenjivač.

Set nepotpune usporedbe postavlja ručno administrator na ekranu „Set up incomplete comparison“. Administrator odabire grupu čiji će se dizajni uspoređivati te dodjeljuje oznaku setu usporedbe. Ukoliko odluči da će set usporedbi biti dodijeljen autorima određene grupe dizajna, mora odabrati opciju „Assign to group“ čime mu se omogućuje odabir grupe koja će obavljati ocjenjivanje, ali mu se ujedno onemogućuje zadavanje broja ocjenjivača jer je on fiksan i jednak broju autora grupe dizajna. U ovom koraku je važno da su svi autori grupe obavili dodavanje

dizajna u sustav kako bi broj ocjenjivača bio ispravno određen te svi dizajni u konačnici bili evaluirani. Administrator može i postaviti vrijednost lambda replikacije (λ) koja označava broj puta koliko će svaki par dizajna biti uspoređen. Što je veća replikacija, veći je i broj usporedbi zadanih ocjenjivačima. Broj usporedbi po ocjenjivaču računa se kao: $\lambda \cdot \frac{n \cdot (n-1)}{2 \cdot m}$, gdje je n broj dizajna u grupi, a m broj ocjenjivača. Broj usporedbi na grupi pomnožen s λ ne mora biti djeljiv s brojem ocjenjivača, što je i dalje prihvatljivo ako se radi u dodijeljenom setu usporedbi. Sustav će se pobrinuti da parovi dizajna predstavljeni kao ostatak pri dijeljenju budu nasumično raspodijeljeni među ocjenjivačima, čime će neki ocjenjivači imati jednu više usporedbu od ostalih, a princip je prikazan na slici 4.18. Ocjenjivači koji imaju dodatnu usporedbu oni su koji se nalaze u prvih p ocjenjivača u polju, gdje je p ostatak pri dijeljenju ukupnog broja usporedbi i broja ocjenjivača, u kodu označen varijablom leftoverPairAmount. Parovi dodijeljeni korisniku selektiraju se LINQ naredbom koja uzima u obzir autore dizajna, replikaciju zbog koje se parovi grupiraju i nasumični odabir.

```
for (int u = 0; u < usersToPreassign.Count; u++)
{
     revisedBlockSize = u < leftoverPairAmount ? blockSize + 1 : blockSize;
     int user = usersToPreassign[u];
     var distinctDesignPairs = pairs
                         .Where(pair => pair.DesignA.AuthorID != user &&
                         pair.DesignB.AuthorID != user)
                         .GroupBy(dp => new { Id1 = Math.Min(dp.DesignA.Id,
                         dp.DesignB.Id), Id2 = Math.Max(dp.DesignA.Id,
                         dp.DesignB.Id) })
                         .Select(group => group.First())
                          .OrderBy(_ => random.Next())
                          .Take(revisedBlockSize)
                          .ToList();
```

Slika 4.18. Selektiranje parova dodijeljenih korisniku

Ako se dogodi slučaj da su preostali nedodijeljeni dizajni samo oni kojima je autor ocjenjivač ili su parovi međusobno identični, algoritam traži kandidata za zamjenu među korisnicima kojima su prethodno dodijeljeni parovi dizajna. Kod traženja kandidata za usporedbu, mora se pronaći prikladan par kod kandidata i među nedodijeljenim parovima. Pronalazak prikladnog para među nedodijeljenim parovima prikazan je kodom na slici 4.19.

Kod nepotpune nedodijeljene usporedbe, broj ocjenjivača odabire administrator, s uvjetom da broj usporedbi po ocjenjivaču bude prirodan broj, što je moguće postići namještanjem lambda replikacije. Dodavanje seta usporedbi inače je onemogućeno, kako se dohvaćanje parova za

usporedbu ne bi dodatno zakompliciralo vođenjem brojača ocjenjivača koji su do sada pristupili procesu usporedbe.

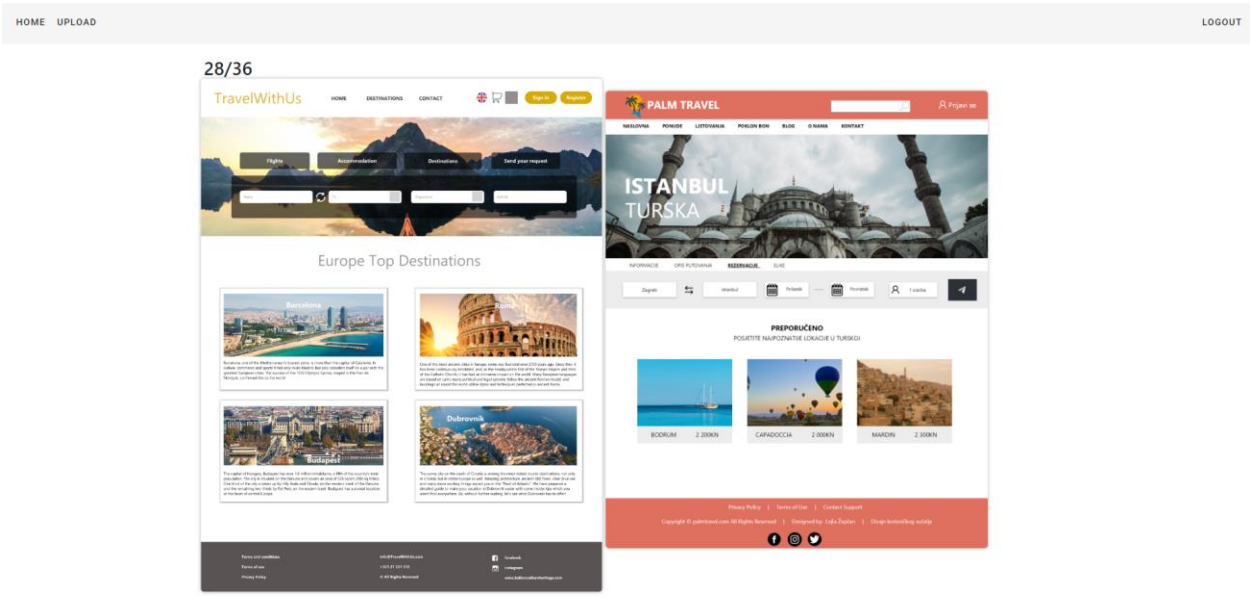
```
return pairs
       .Where(pair =>
               pair.DesignA.AuthorID != resultTuple?.Item1 && //find a pair
           of designs not authored by the swap candidate
               pair.DesignB.AuthorID != resultTuple?.Item1 &&
               !assignedComparisons.Any(tuple =>  //the swap candidate
           cannot  have that exact pair already assigned to them
                  tuple.Item1 == resultTuple?.Item1 &&
                 (tuple.Item2.DesignA.Id == pair.DesignA.Id &&
                  tuple.Item2.DesignB.Id == pair.DesignB.Id)))
        .FirstOrDefault();
```

Slika 4.19. Odabir para koji će se dati kandidatu za zamjenu

4.4.6. Uspoređivanje dizajna

Kako bi korisnik započeo usporedbu dizajna, mora znati oznaku i vrstu seta usporedbi. U slučaju seta potpunih usporedbi, sustav generira parove dizajna kojima korisnik nije autor i šalje ih u odgovoru. Kod seta potpunih dodijeljenih usporedbi, sustav vraća samo parove koji su unaprijed dodijeljeni prijavljenom korisniku, a kod nedodijeljenih usporedbi vraća b parova kojima nije autor trenutni korisnik, gdje je b veličina bloka, odnosno broj usporedbi po korisniku. Sustav odmah u bazi označava parove usporedbi kao dodijeljene trenutnom korisniku, tako da korisnik opet dobije iste parove ako se odjavi i ponovno prijavi u sustav. Ukoliko su svi parovi već dodijeljeni, sustav ne šalje parove, nego odgovarajuću poruku. Ukoliko se dogodi slučaj spomenut u prethodnom potpoglavlju - da nema prikladnih parova za korisnika, potrebno je promijeniti ocjenjivača. Iz tog razloga, nepotpune nedodijeljene usporedbe nisu namijenjene grupama ocjenjivača koji su autori radova te iste grupe, nego im je primarna namjena omogućavanje nepotpunih usporedbi korisnicima koji uopće nisu autori dizajna u sustavu ili pripadaju različitim grupama. Optimalna opcija je raditi potpunu ili nepotpunu dodijeljenu usporedbu kada god je to moguće.

Korisnik može pauzirati uspoređivanje, kao i ocjenjivanje, dokle god je prijavljen, a prije postavljanja rezultata u bazu podataka jer se uspoređeni parovi spremaju u lokalnu pohranu. Nedovršeni set usporedbi nije moguće postaviti u bazu podataka te u slučaju odjave iz sustava, usporedba se vrši ispočetka. Izgled sučelja tijekom obavljanja usporedbe prikazan je slikom 4.20.



Slika 4.20. Uspoređivanje seta dizajna

4.4.7. Rangiranje dizajna

Administrator ima uvid u rezultate uspoređivanja dizajna, odnosno kalkulirani rang pojedinog dizajna u grupi. Dizajni se rangiraju po postotku pobjeda u usporedbama, a ukoliko dva ili više ranga imaju jednak postotak, gleda se koji se pokazao boljim u međusobnim direktnim usporedbama. Za ostvarivanje takvog rangiranja, implementiran je prilagođeni uspoređivač (engl. *comparer*), čija je implementacija sučelja IComparer prikazana slikom 4.21. Ovakav sustav rangiranja valjan je zato što je lambda replikacija konstantna, odnosno svaki par usporedbi ukupno se pojavio λ puta, pri čemu je λ=n kod potpune usporedbe, odnosno λ=n-1 kada svaki ocjenjivač ima točno 1 svoj dizajn dodan u grupu. Iz tog razloga, administrator bi trebao izbjegavati dodavanje više dizajna istog autora u grupu ako će taj isti autor ocjenjivati dizajne te grupe. Ako autori nemaju jednak broj dizajna u grupi ili ako su pojedini ocjenjivači autori, a drugi nisu, λ prestaje biti konstantna te to može utjecati na točnost rezultata.

```
public int Compare(DesignComparisonsDTO a, DesignComparisonsDTO b)
        {
            var winPercentageA = a.WinCount * 1.0 / a.ComparisonCount;
            var winPercentageB = b.WinCount * 1.0 / b.ComparisonCount;
            if (winPercentageA > winPercentageB)
            {
                    return 1;
            }
            if (winPercentageA < winPercentageB)
            {
                    return -1;
            }
            else
            {
                    var faceToFaceComparisons = context.Comparison.Where(c =>
                    c.DesignAid == a.DesignId && c.DesignBid == b.DesignId ||
                    c.DesignAid == b.DesignId && c.DesignBid == a.DesignId);
                    var winCountA = faceToFaceComparisons.Count(c =>
                    c.WinningDesignId == a.DesignId);
                    var winCountB = faceToFaceComparisons.Count(c =>
                    c.WinningDesignId == b.DesignId);
                    return winCountA > winCountB ? 1 : -1;
            }
        }
```

Slika 4.21. Implementacija metode Compare prilagođenog uspoređivača dizajna

4.4.8. Provjera statusa evaluacije

Administrator može i provjeriti statuse ocjenjivanja i uspoređivanja ocjenjivača, odnosno vidjeti koji korisnik je što odradio. Statusi ocjenjivanja pregledavaju se za danu grupu, a statusi uspoređivanja za dani set usporedbi. Kod ocjenjivanja, pretpostavlja se da će ocjenjivanje izvršiti autori te iste grupe, te su prikazani njihovi statusi evaluacije, a isto vrijedi i za potpunu usporedbu. Kod nedodijeljene nepotpune usporedbe, prikazuju se statusi evaluacije za trenutno dodijeljene ocjenjivače i koliko ocjenjivača je još potrebno za dovršetak evaluacije (slika 4.22.). Kod dodijeljene nepotpune usporedbe, prikazuju se statusi uspoređivanja za sve korisnike kojima je dodijeljeno uspoređivanja tog seta usporedbi.

Slika 4.22. Status evaluacije za nedodijeljenu nepotpunu usporedbu

4.4.9. Izvoz rezultata

Rezultati evaluacije svake grupe mogu se osim pregleda kroz aplikaciju i izvesti u XLSX formatu. Excel datoteka sadrži listove s ocjenama, rangovima za svaki set usporedbi i matricama parnog uspoređivanja za svaki set usporedbi (slika 4.23.). Time se omogućava daljnja obrada podataka kao i vizualni prikaz međusobnih direktnih usporedbi dizajna.

	A	B	C	D	E	F	G	H	I	J	K
1		**85**	**86**	**87**	**88**	**89**	**90**	**91**	**92**	**93**	**94**
2	**85**	-	15	15	15	14	3	14	2	14	15
3	**86**	0	-	11	7	12	0	9	0	11	15
4	**87**	0	4	-	6	11	0	7	0	10	14
5	**88**	0	8	9	-	12	0	6	0	10	14
6	**89**	1	3	4	3	-	1	4	1	7	13
7	**90**	12	15	15	15	14	-	14	9	14	15
8	**91**	1	6	8	9	11	1	-	2	12	15
9	**92**	13	15	15	15	14	6	13	-	15	15
10	**93**	1	4	5	5	8	1	3	0	-	15
11	**94**	0	0	1	1	2	0	0	0	0	-
12											

Slika 4.23. Generirana Excel tablica

5. EKSPERIMENT

Kako bi se provjerilo jesu li studenti dovoljno objektivni da bi evaluirali tuđe radove, proveden je eksperiment u tri dijela. Grupa ispitanika sačinjena je od 15 studenata FERIT-a upisanih na Fakultet tijekom vremena izvođenja eksperimenta.

5.1. Tijek eksperimenta

Ulazne podatke sačinjavalo je 10 studentskih radova – dizajna grafičkih korisničkih sučelja koje su studenti 2022. godine dizajnirali za potrebe kolegija na FERIT-u. Zadatak je bio dizajnirati naslovnu stranicu fiktivne turističke agencije. Iako radove u stvarnoj situaciji studenti mogu predati ponaosob, za potrebe eksperimenta stvoreni su izmišljeni korisnički računi i aliasi za svakog od 10 autora te su radovi podneseni preko administratorskog sučelja.

Eksperiment je sačinjen od 3 dijela. Prvi dio kardinalno je ocjenjivanje radova ocjenama od 1 do 5, gdje 1 predstavlja ocjenu nedovoljan, a 5 ocjenu odličan. Osim studenata, radove je za referencu ocijenio i jedan od nastavnika. Drugi dio ordinalno je ocjenjivanje metodom nepotpune nedodijeljene usporedbe. Lambda replikacija postavljena je na 3, čime je veličina bloka, odnosno količina usporedbi koju obavlja svaki ocjenjivač, jednaka 9. Treći dio ordinalno je ocjenjivanje metodom potpune usporedbe, gdje je svaki ocjenjivač obavio svih 45 usporedbi.

Izlazni podaci prvog dijela ocjene su od 1 do 5, odnosno od nedovoljan do odličan, a drugog i trećeg dijela matrice parne usporedbe pretvorene u rangove dizajna na temelju postotka pobjede u usporedbama. Sljedeća potpoglavlja opisuju i analiziraju rezultate eksperimenta.

5.2. Usporedba ocjena ispitanika i nastavnika

Ocjene ispitanika (slika 5.1.) uspoređene su s prethodno dodijeljenim ocjenama nastavnika (tablica 5.1.).

Tablica 5.1. Ocjene nastavnika dodijeljene aliasima autora

Ocjene nastavnika	
Prvi Student	5
Testko Testic	5
Testko Kovic	4.5
Dora Doric	3.5
Drugi Student	4.5
Novi Three	4
Pero Perić	3
Ivo Ivic	4
Petra Perić	3
Marko Markić	2

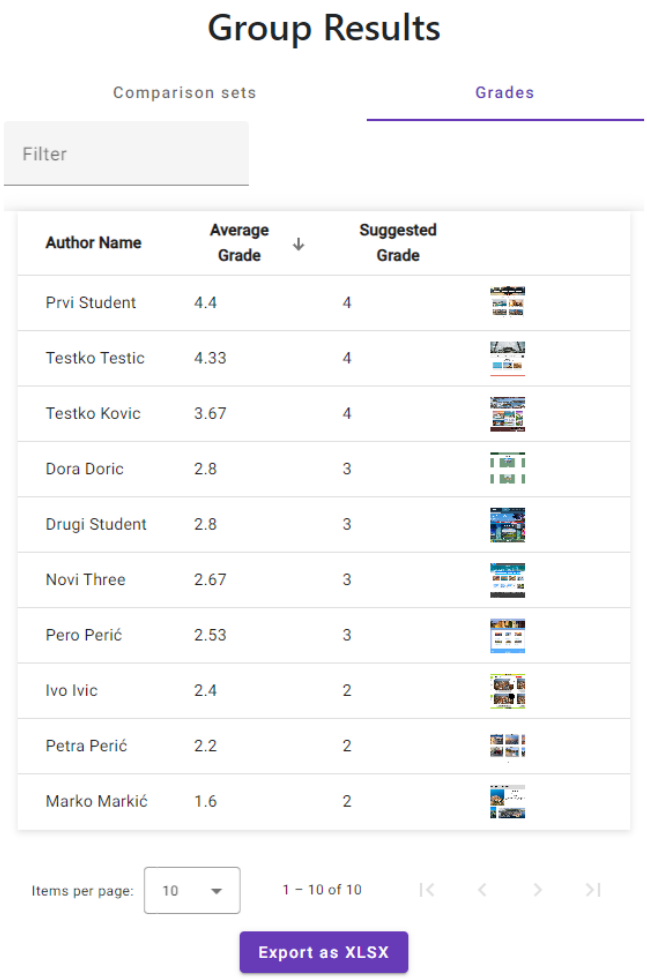
Za mjerenje korelacije korišten je Pearsonov koeficijent korelacije. Pearsonov koeficijent koristi se uz pretpostavku da su varijable X i Y linearno povezane, odnosno da između njih postoji neprekidna normalna distribucija. Pearsonov koeficijent omjer je kovarijance i standardne devijacije i prikazan je jednadžbom (5-1) [34].

$$r_{XY} = \frac{cov(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{5-1}$$

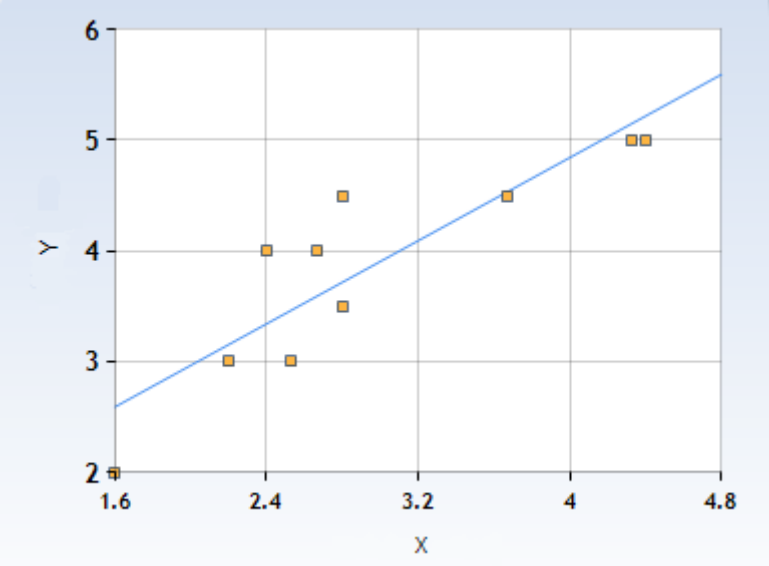
Granične vrijednosti pri ocjenjivanju vrijednosti koeficijenta su:

- 0.00 - 0.19 (vrlo slaba korelacija)
- 0.20 - 0.39 (slaba korelacija)
- 0.40 - 0.59 (srednja korelacija)
- 0.60 - 0.79 (snažna korelacija)
- 0.80 - 1.00 (vrlo snažna korelacija).

U eksperimentu varijabla X prosječna je ocjena ispitanika, a varijabla Y ocjena nastavnika. Uvrštavanjem vrijednosti u danu formulu, Pearsonov koeficijent iznosi 0.8804, što ukazuje na vrlo snažnu korelaciju. Dijagram raspršenja prikazan je slikom 5.2.



Slika 5.1. Ocjene ispitivača dodijeljene aliasima autora



Slika 5.2. Dijagram raspršenja

5.3. Usporedba rangova ispitanika i nastavnika

Potrebno je utvrditi stupanj korelacije između rangova ispitanika i rangova nastavnika kako bi se provjerilo jesu li studenti objektivni evaluatori kada koriste metodu potpune usporedbe. Drugi stupac tablice 5.2. prikazuje poredak koji je dodijelio nastavnik, a treći stupac poredak dobiven usporedbama ispitanika metodom potpune usporedbe. Kompletna statistika potpune usporedbe prikazana je slikom 5.3.

Tablica 5.2. Rangovi dobiveni iz pojedine metode evaluacije

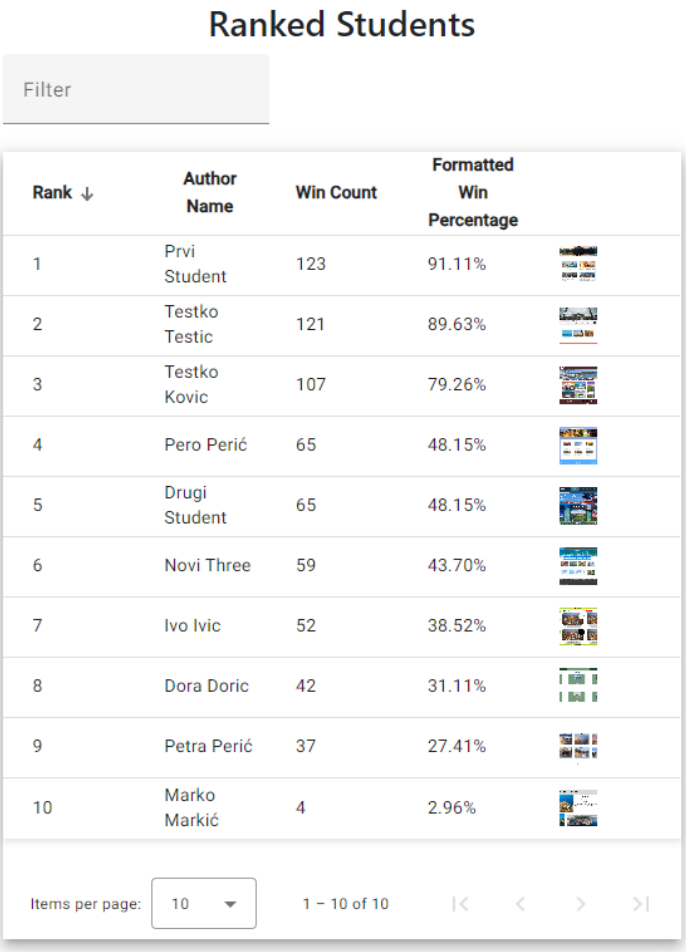
Ocjene ispitanika	Rang nastavnika	Potpuna usporedba	Nepotpuna usporedba
1. Prvi Student	1. Testko Testic	1. Prvi Student	1. Prvi Student
2. Testko Testic	2. Prvi Student	2. Testko Testic	2. Testko Testic
3. Testko Kovic	3. Drugi Student	3. Testko Kovic	3. Testko Kovic
4. Drugi Student	4. Testko Kovic	4. Pero Perić	4. Pero Perić
4. Dora Doric	5. Novi Three	5. Drugi Student	5. Novi Three
6. Novi Three	6. Ivo Ivic	6. Novi Three	6. Drugi Student
7. Pero Perić	7. Dora Doric	7. Ivo Ivic	7. Ivo Ivic
8. Ivo Ivic	8. Pero Perić	8. Dora Doric	8. Dora Doric
9. Petra Perić	9. Petra Perić	9. Petra Perić	9. Petra Perić
10. Marko Markić	10. Marko Markić	10. Marko Markić	10. Marko Markić

Za određivanje korelacije između rangova korišten je Spearmanov koeficijent korelacije koji ima za cilj izmjeriti dosljednost povezanosti u poretku varijabli. Može se koristiti za mjerenje jačine korelacije bilo koje monotone povezanosti, bila ona linearna ili ne. Granične vrijednosti pri ocjenjivanju vrijednosti koeficijenta jednake su kao kod Pearsonovog koeficijenta. Spearmanov koeficijent korelacije, često označen kao Spearmanov ro (ρ) računa se na način prikazan jednadžbom (5-2), gdje d_i označava razliku između rangova istog dizajna, a n broj dizajna koji se uspoređuju [35]. Spearmanov ro iznosi 0.8424 čime se može zaključiti da postoji vrlo snažna korelacija između rangova dobivenih iz usporedbi ispitanika i rangova nastavnika. Drugi koeficijent korelacije između rangova je Kendallov tau (τ), koji se također može koristiti za mjerenje jačine korelacije bilo koje monotone povezanosti. Kendallov koeficijent korelacije računa se na način prikazan jednadžbom (5-3), gdje je n_c broj suglasnih parova, a n_d broj nesuglasnih parova. Par promatranja (x_i,y_i) i (x_j,y_j) suglasan je ako se rangovi međusobno slažu, tj. ako je $x_i > x_j$ i $y_i > y_j$ ili $x_i < x_j$ i $y_i < y_j$. Par je nesuglasan ako $x_i > x_j$ and $y_i < y_j$ ili $x_i < x_j$ i $y_i > y_j$.

Ako vrijedi $x_i = x_j$ i/ili $y_i = y_j$ par nije niti suglasan niti nesuglasan. U slučaju rangova izvedenih iz usporedbi ispitanika i rangova nastavnika, broj suglasnih parova je 39, a nesuglasnih 6. Time je Kendallov koeficijent korelacije jednak 0.7333 što ukazuje na snažnu korelaciju [36].

$$\rho = 1 - \frac{6 \cdot \Sigma\, d_i}{n \cdot (n^2 - 1)} \tag{5-2}$$

$$\tau = \frac{n_c - n_d}{n_c + n_d} \tag{5-3}$$



Slika 5.3. Poredak i statistika dobiveni usporedbama ispitanika metodom potpune usporedbe

5.4. Usporedba ocjena i rangova

Sljedeći korak u analizi rezultata određivanje je korelacije između ocjena i rangova ispitanika. Kako bi kardinalno i ordinalno ocjenjivanje bilo normalizirano, ocjene su svedene na rangove. Potrebno je odrediti korelaciju između rangova dobivenih iz ocjena ispitanika i rangova dobivenih iz usporedbi ispitanika. Prvi stupac tablice 5.2. prikazuje rangove dobivene iz ocjena ispitanika, a treći stupac poredak dobiven usporedbama ispitanika metodom potpune usporedbe.

Spearmanov ro iznosi 0.8632 čime se može zaključiti da postoji vrlo snažna korelacija između rangova sintetiziranih iz ocjena ispitanika i rangova iz usporedbi ispitanika. U slučaju rangova izvedenih iz ocjena ispitanika i usporedbi ispitanika, broj suglasnih parova je 39, a nesuglasnih 5. Time je Kendallov koeficijent korelacije jednak 0.7727 što ukazuje na snažnu korelaciju.

5.5. Usporedba rangova potpune i nepotpune usporedbe

Kod velikog broja radova, broj usporedbi raste eksponencijalno, što dovodi do nemogućnosti provođenja potpune usporedbe zbog zamora ispitanika. Potrebno je provjeriti je li nepotpuna usporedba dovoljno dobra da zamijeni potpunu usporedbu kada je nužno. S obzirom na to da su isti ispitanici izvršili potpunu i nepotpunu usporedbu nad istim setom dizajna, moguće je odrediti Spearmanov i Kendallov koeficijent korelacije između poretka dobivenog usporedbama metodom potpune parne usporedbe i metodom nepotpune parne usporedbe. Rangovi dobiveni metodom nepotpune parne usporedbe prikazani su četvrtim stupcem tablice 5.2., a potpuna statistika slikom 5.4.

Budući da je poredak nepotpunom usporedbom gotovo identičan onom dobivenom popunom usporedbom, izuzev dizajna na 5. i 6. mjestu, Spearmanov ro vrlo je visok te iznosi 0.9879 što ukazuje na vrlo snažnu korelaciju. Kod izračuna Kendallovog koeficijenta, broj suglasnih parova je 44, a nesuglasnih 1, čime je on jednak 0.9556, također ukazujući na vrlo snažnu korelaciju.

Za dane parametre metoda nepotpune usporedbe pokazala se zadovoljavajućom zamjenom u slučaju većeg broja dizajna.

Rank ↓	Author Name	Win Count	Formatted Win Percentage	
1	Prvi Student	26	96.30%	
2	Testko Testic	23	85.19%	
3	Testko Kovic	20	74.07%	
4	Pero Perić	15	55.56%	
5	Novi Three	14	51.85%	
6	Drugi Student	10	37.04%	
7	Ivo Ivic	9	33.33%	
8	Dora Doric	9	33.33%	
9	Petra Perić	6	22.22%	
10	Marko Markić	3	11.11%	

Items per page: 10 1 – 10 of 10

Slika 5.4. Poredak i statistika dobiveni usporedbama ispitanika metodom nepotpune usporedbe

5.6. Utjecaj smanjenja lambda replikacije

U slučajevima kada je broj ocjenjivača nizak, moguće je da će kod nepotpune usporedbe biti potrebno lambda replikaciju postaviti na nižu vrijednost. U svrhu testiranja rezultata, simulirano je smanjenje lambda replikacije na setu podataka dobivenom eksperimentom. Za simulaciju smanjenja lambde napravljena je izmjena u kodu prikazana slikom 5.5.

```
Random random = new Random();

var comparisons = context.Comparison
                .Where(c => c.ComparisonSetId == id).ToList()
                .GroupBy(c => new {
                                DesignAid = c.DesignAid,
                                DesignBid = c.DesignBid })
                .SelectMany(group =>
                       group.OrderBy(_ => random.Next()).Take(lambda)).ToList();
```

Slika 5.5. Izmjena u kodu za simulaciju smanjenja lambda replikacije

5.6.1. Smanjenje lambda replikacije na 2

Nasumice je odabrano 2 od 3 obavljene usporedbe za svaki par dizajna, čime je lambda replikacija očuvana konstantnom. Ukupan broj obavljenih usporedbi u tom slučaju iznosi 90. Dobiveni poredak prikazan je slikom 5.6.

Kada se izračunaju koeficijenti korelacije između poretka potpune usporedbe i poretka dobivenog simulacijom, Spearmanov ro jednak je 0.9515, a Kendallov tau 0.8667. Oba i dalje ukazuju na vrlo snažnu korelaciju, iako su nešto niži od koeficijenata dobivenih za λ=3.

Rank ↓	Author Name	Win Count	Formatted Win Percentage	
1	Prvi Student	17	94.44%	
2	Testko Testic	16	88.89%	
3	Testko Kovic	13	72.22%	
4	Novi Three	10	55.56%	
5	Pero Perić	9	50.00%	
6	Drugi Student	7	38.89%	
7	Dora Doric	7	38.89%	
8	Ivo Ivic	6	33.33%	
9	Petra Perić	4	22.22%	
10	Marko Markić	1	5.56%	

Items per page: 10 1 – 10 of 10

Slika 5.6. Poredak dizajna za λ=2

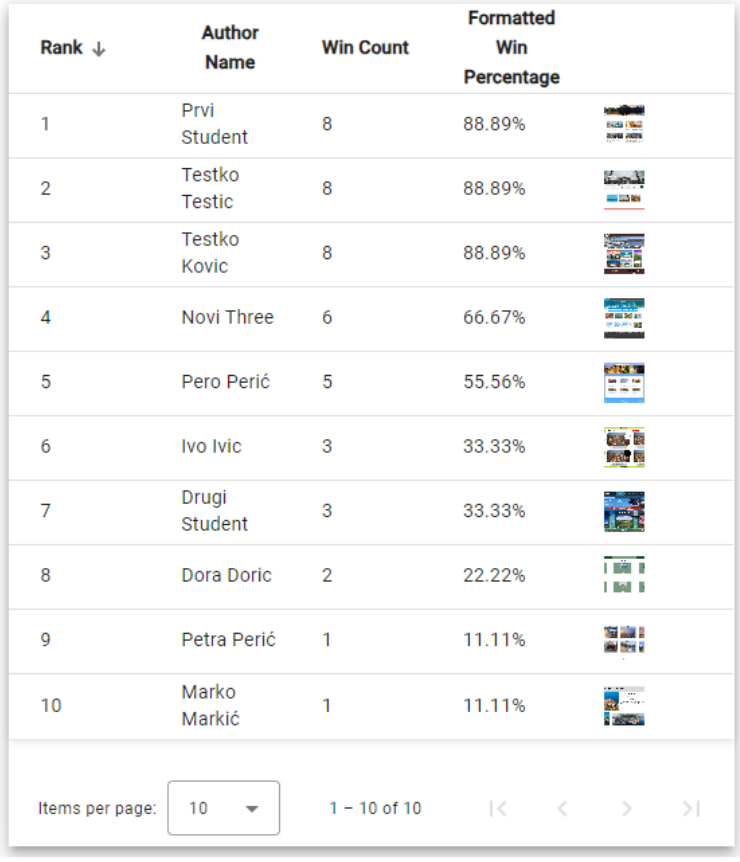
5.6.2. Smanjenje lambda replikacije na 1

Za slučaj kada je simulirano postavljanje lambda replikacije na 1, ukupan broj usporedbi jednak je 45, a dobiveni poredak prikazuje slika 5.7. Koeficijenti korelacije između poretka potpune usporedbe i poretka dobivenog simulacijom za λ=1 su slijedeći: Spearmanov ro nešto je manji nego u prethodnom slučaju i iznosi 0.9394, a Kendallov tau ostao je isti, tj. 0.8667. Smanjenjem

lambda replikacije vidljiv je trend smanjenja korelacije s rezultatima potpune usporedbe, kako je prikazano tablicom 5.3., no koeficijenti i dalje ukazuju na vrlo snažnu korelaciju.

Tablica 5.3. Pregled koeficijenata korelacije

	Rangovi nastavnika i rangovi studenata	**Ocjene i rangovi potpune usporedbe**	**Rangovi potpune i nepotpune usporedbe (λ=3)**	**Rangovi potpune i nepotpune usporedbe (λ=2)**	**Rangovi potpune i nepotpune usporedbe (λ=1)**
Spearmanov ro	0.8424	0.8632	0.9879	0.9515	0.9394
Kendallov tau	0.7333	0.7727	0.9556	0.8667	0.8667



Rank ↓	Author Name	Win Count	Formatted Win Percentage	
1	Prvi Student	8	88.89%	
2	Testko Testic	8	88.89%	
3	Testko Kovic	8	88.89%	
4	Novi Three	6	66.67%	
5	Pero Perić	5	55.56%	
6	Ivo Ivic	3	33.33%	
7	Drugi Student	3	33.33%	
8	Dora Doric	2	22.22%	
9	Petra Perić	1	11.11%	
10	Marko Markić	1	11.11%	

Items per page: 10 1 – 10 of 10

Slika 5.7. Poredak dizajna za λ=1

6. ZAKLJUČAK

Zadatak diplomskog rada bio je razviti web sučelje koji omogućava kardinalno te ordinalno ocjenjivanje dizajna grafičkih korisničkih sučelja među vršnjacima. Zatim je bilo potrebno provesti eksperiment kojim bi se istražilo jesu li studenti dovoljno objektivni ocjenjivači.

Razvijeno je web sučelje koje omogućava dodavanje dizajna u bazu podataka te pregled i brisanje dizajna po grupama studenata. Administratoru je omogućeno pregledavanje statusa evaluacije radova. Uz to, u svrhu smanjenja opterećenja ocjenjivačima, administrator može postaviti ocjenjivanje nepotpunom metodom usporedbe te dodijeliti zadatke ocjenjivanja pojedinoj skupini studenata. Administratoru je omogućen izračun prosječne ocjene dizajna te rangiranje dizajna unutar jedne skupine na temelju evaluacije parnom usporedbom. Rezultate može i izvesti u Excel proračunsku tablicu. Proveden je eksperiment nad 15 ispitanika – studenata FERIT-a u tri dijela kako bi se evaluirale metode kardinalnog i ordinalnog ocjenjivanja studentskih radova. Testirana je i korelacija rezultata uz smanjeni broj parnih usporedbi.

Rezultati eksperimenta pokazali su da su prosječne ocjene ispitanika vrlo snažno korelirane s ocjenama nastavnika te da su rangovi ispitanika vrlo snažno korelirani s rangovima nastavnika. Metoda parne usporedbe pokazala se dostojnom zamjenom za kardinalno ocjenjivanje, pritom smanjujući pristranost. Uspoređene su metode potpune i nepotpune usporedbe te je između njih ustanovljena vrlo snažna korelacija što je pozitivni iskorak ka smanjenju opterećenja na ocjenjivače kod velikih setova dizajna. Simulacijom je testiran utjecaj lambda replikacije kod nepotpune usporedbe na rezultate te je detektiran trend osnaženja korelacije s rezultatima potpune usporedbe povećanjem lambda replikacije.

Moguće je proširiti istraživanje proučavanjem rezultata ocjenjivanja kod promjenjive lambda replikacije i provjeriti jesu li rangovi dobiveni tom metodom i dalje snažno korelirani s onima dobivenim metodom potpune usporedbe. Dodatan korak bio bi treniranje modela umjetne inteligencije uz pomoć podataka skupljenih ocjenjivanjem vršnjaka čime bi se dodatno umanjila subjektivnost ocjenjivača.

LITERATURA

[1] E., Popkova, „QUALITY OF DIGITAL PRODUCT: THEORY AND PRACTICE“, *Int. J. Qual. Res.*, sv. 14, str. 201–218, ožu. 2020.

[2] N., Hemidli, „Introduction to UI/UX Design: Key Concepts and Principles“, str. 1–4, ožu. 2023.

[3] A., Kahng, Y., Kotturi, C., Kulkarni, D., Kurokawa, A., Procaccia, „Ranking Wily People Who Rank Each Other“, *Proc. AAAI Conf. Artif. Intell.*, izd. 1, sv. 32, tra. 2018.

[4] J., Cambre, S., Klemmer, C., Kulkarni, „Juxtapeer: Comparative Peer Review Yields Higher Quality Feedback and Promotes Deeper Reflection“, u *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, str. 1–13, New York, NY, USA, 2018.

[5] N. B., Shah, M. J., Wainwright, „Simple, robust and optimal ranking from pairwise comparisons“, *J. Mach. Learn. Res.*, izd. 1, sv. 18, str. 7246–7283, sij. 2017.

[6] R., Heckel, M., Simchowitz, K., Ramchandran, M., Wainwright, „Approximate Ranking from Pairwise Comparisons“, u *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, str. 1057–1066, 2018.

[7] E., Agbozo, „A Hybrid Data-Driven Web-Based UI-UX Assessment Model“. arXiv, 21-sij-2023.

[8] H., Joo, „A Study on Understanding of UI and UX, and Understanding of Design According to User Interface Change“, izd. 20, sv. 12, 2017.

[9] T., Bizimungu, D., Harelimana, J., Marie Ntaganda, „A Client-Server and Web-Based Graphical User Interface Design for the Mathematical Model of Cardiovascular-Respiratory System“, *Appl. Comput. Intell. Soft Comput.*, sv. 2021, str. e5581937, svi. 2021.

[10] A. H., Allam, A. R. C., Hussin, H. M., Dahlan, „User Experience: Challenges and Opportunities“.

[11] K., Rodden, H., Hutchinson, X., Fu, „Measuring the User Experience on a Large Scale: User-Centered Metrics for Web Applications“.

[12] W., Xinghai, „Conceptualizing Synergies Between SEO Practices and UX Design for Enhanced Website Usability and Engagement“, *J. Digit. Realism Mastery DREAM*, sv. 2, str. 18–22, svi. 2023.

[13] „From UX to CX: Rethinking the Digital User Experience as a Collaborative Exchange“, *MIT Initiat. Digit. Econ.*, svi. 2017.

[14] E. L.-C., Law, V., Roto, M., Hassenzahl, A. P. O. S., Vermeeren, J., Kort, „Understanding, scoping and defining user experience“, *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2009.

[15] P., Marti, I., Iacono, „Anticipated, Momentary, Episodic, Remembered: the many facets of User eXperience“, predstavljeno na 2016 Federated Conference on Computer Science and Information Systems, str. 1647–1655, 2016.

[16] A., Vermeeren, V., Roto, J., Kort, „User Experience White Paper: Bringing clarity to the concept of user experience“, velj. 2011.

[17] UW-La Crosse, „Evaluating User Interfaces“, User Interface Design, 2015.

[18] Y., Ghanam, „Discount Usability Testing“, *Department of Computer Science, University of Calgary*, pros. 2007.

[19] G., Meiselwitz, B., Wentz, J., Lazar, „Universal Usability: Past, Present, and Future“, *Found. Trends® Human–Computer Interact.*, izd. 4, sv. 3, str. 213–333, kol. 2010.

[20] N., Bevan, „20 Chapter - Cost-Benefit Framework and Case Studies“, u *Cost-Justifying Usability (Second Edition)*, R. G. Bias i D. J. Mayhew, Ur. Morgan Kaufmann: San Francisco, 2005, str. 575–600.

[21] K., Romić, H., Leventić, L., Mandić, I., Galić, „Using Peer Grading to Improve the Objectivity of Grades in GUI Design Courses“, u *2023 International Symposium ELMAR*, str. 181–184, 2023.

[22] H., Luo, A. C., Robinson, J.-Y., Park, „Peer Grading in a MOOC: Reliability, Validity, and Perceived Effects“, *Online Learn.*, izd. 2, sv. 18, lip. 2014.

[23] K., Raman, T., Joachims, „Methods for Ordinal Peer Grading“, *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, tra. 2014.

[24] C., Kamolsin, F., Pensiri, K. H., Ryu, P., Visutsak, „The Evaluation of GUI Design using Questionnaire and Multivariate Testing“, u *2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, str. 191–195, 2022.

[25] G., Liebel, „Structuring GUI Acceptance Tests based on Usage Scenarios“, 2013.

[26] gewarren, .„NET (and .NET Core) - introduction and overview“ [online], 24-ožu-2023. Dostupno na: https://learn.microsoft.com/en-us/dotnet/core/introduction. [Pristupljeno: 24.6.2023.].

[27] BillWagner, „A tour of C# - Overview“ [online], 04-svi-2023. Dostupno na: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/. [Pristupljeno: 24.6.2023.].

[28] „Angular - What is Angular?“ [online]. Dostupno na: https://angular.io/guide/what-is-angular. [Pristupljeno: 24.6.2023.].

[29] WilliamDAssafMSFT, „What is the Azure SQL Database service? - Azure SQL Database“ [online], 03-ožu-2023. Dostupno na: https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview. [Pristupljeno: 24.6.2023.].

[30] markingmyname, „Use Schema Compare to Compare Different Database Definitions - SQL Server Data Tools (SSDT)“ [online], 20-lip-2023. Dostupno na: https://learn.microsoft.com/en-us/sql/ssdt/how-to-use-schema-compare-to-compare-different-database-definitions?view=sql-server-ver16. [Pristupljeno: 14.1.2024.].

[31] IEvangelist, „Dependency injection - .NET“ [online], 15-pros-2023. Dostupno na: https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection. [Pristupljeno: 16.1.2024.].

[32] Rick-Anderson, „Model Binding in ASP.NET Core“ [online], 22-lip-2023. Dostupno na: https://learn.microsoft.com/en-us/aspnet/core/mvc/models/model-binding?view=aspnetcore-8.0. [Pristupljeno: 16.1.2024.].

[33] „Angular - Workspace and project file structure“ [online]. Dostupno na: https://angular.io/guide/file-structure. [Pristupljeno: 16.1.2024.].

[34] „Pearson Correlation Coefficient - an overview | ScienceDirect Topics“ [online]. Dostupno na: https://www.sciencedirect.com/topics/social-sciences/pearson-correlation-coefficient. [Pristupljeno: 30.1.2024.].

[35] „Spearman Rank - an overview | ScienceDirect Topics“ [online]. Dostupno na: https://www.sciencedirect.com/topics/engineering/spearman-rank. [Pristupljeno: 30.1.2024.].

[36] M.-T., Puth, M., Neuhäuser, G. D., Ruxton, „Effective use of Spearman’s and Kendall’s correlation coefficients for association between two measured traits“, *Anim. Behav.*, sv. 102, str. 77–84, tra. 2015.

POPIS KRATICA

API	Application Programming Interface
BM	Binding Model
CLI	Command Line Interface
CLR	Common Language Runtime
CUI	Character User Interface
DAL	Data Access Layer
DTO	Dana Transfer Object
ER	Entity Relationship
FERIT	Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
GUI	Graphical User Interface
HCI	human-computer interaction
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JWT	JSON Web Token
LINQ	Language-Integraded Query
MOOC	massive open online course
NFT	non-fungible token
npm	node package manager
PWC	pairwise comparison
SEO	search engine optimization
SPA	single-page application
SQL	Structured Query Language
UX	User eXperience

POPIS SLIKA

Slika 3.1. Vremenska lenta korisničkog iskustva 7
Slika 4.1. Usporedba sintakse LINQ 12
Slika 4.2. Slojevi programskog sustava 13
Slika 4.3. Dijagram entiteta i veza 14
Slika 4.4. SQL skripta za stvaranje tablice 15
Slika 4.5. Spajanje na bazu podataka 16
Slika 4.6. Automatski generirana klasa entiteta 16
Slika 4.7. Model vezivanja za akademsku godinu 17
Slika 4.8. Mapiranje modela entiteta na DTO 17
Slika 4.9. Mapiranje modela vezivanja na model entiteta 17
Slika 4.10. Javna metoda servisa AcademicYearService 17
Slika 4.11. Ubrizgavanje ovisnosti 18
Slika 4.12. POST metoda za dodavanje akademske godine 18
Slika 4.13. Dodavanje autorizacijskog zaglavlja HTTP zahtjevu 20
Slika 4.14. Postupak prijave na klijentskoj aplikaciji 21
Slika 4.15. Dodavanje grupe 22
Slika 4.16. Pregled dizajna grupe 23
Slika 4.17. Ocjenjivanje grupe dizajna 24
Slika 4.18. Selektiranje parova dodijeljenih korisniku 25
Slika 4.19. Odabir para koji će se dati kandidatu za zamjenu 26
Slika 4.20. Uspoređivanje seta dizajna 27
Slika 4.21. Implementacija metode Compare prilagođenog uspoređivača dizajna 28
Slika 4.22. Status evaluacije za nedodijeljenu nepotpunu usporedbu 29
Slika 4.23. Generirana Excel tablica 29
Slika 5.1. Ocjene ispitivača dodijeljene aliasima autora 32
Slika 5.2. Dijagram raspršenja 32
Slika 5.3. Poredak dobiven usporedbama ispitanika metodom potpune usporedbe 34
Slika 5.4. Poredak dobiven usporedbama ispitanika metodom nepotpune usporedbe 35
Slika 5.5. Izmjena u kodu za simulaciju smanjenja lambda replikacije 36
Slika 5.6. Poredak dizajna za λ=2 37
Slika 5.7. Poredak dizajna za λ=1 38

SAŽETAK

Ovaj rad bavi se istraživanjem objektivnosti studenata pri međusobnom ocjenjivanju radova iz područja dizajna grafičkih korisničkih sučelja. U tu svrhu kreirana je web aplikacija koja omogućava kardinalno i ordinalno ocjenjivanje radova prijavljenim korisnicima. Proveden je eksperiment s 15 ispitanika – studenata FERIT-a koji su ocjenjivali stvarne radove drugih studenata. Ispitanici su morali ocijeniti radove ocjenama od 1 do 5, a potom i evaluirati dizajne metodom potpune i nepotpune parne usporedbe. Rezultati parne usporedbe korišteni su za stvaranje rang liste studentskih radova. Ispitana je korelacija ocjena studenata i nastavnika koristeći Pearsonov koeficijent korelacije. Korelacija rangova studenata i nastavnika te korelacija dvije metode parne usporedbe ispitane su koristeći Spearmanov i Kendallov koeficijent korelacije. Promotrena je i povezanost između kardinalnog i ordinalnog ocjenjivanja te utjecaj smanjenja lambda replikacije. Rezultati su pokazali da je parna usporedba prikladan način ocjenjivanja vršnjaka te da je moguće smanjiti opterećenje na ocjenjivače metodom nepotpune usporedbe i pritom zadržati visoku razinu korelacije s rezultatima potpune usporedbe.

Ključne riječi: dizajn grafičkog korisničkog sučelja, kardinalno i ordinalno ocjenjivanje, lambda replikacija, ocjenjivanje vršnjaka, parna usporedba

ABSTRACT

Development of web environment for grading and comparison of graphic user interfaces

This paper focuses on researching the objectivity of students in evaluating each other's assignments in the field of graphical user interface design. To conduct this research, a web application was created that allowed registered users to evaluate assignments using both cardinal and ordinal evaluation. The experiment involved 15 FERIT students who evaluated the real assignments of other students. The respondents assessed the assignments using marks from 1 to 5 and then evaluated the designs using complete and incomplete pairwise comparison methods. The results of the pairwise comparison were then used to create a ranking list of student papers. The Pearson correlation coefficient was used to determine the correlation between the grades given by students and a teacher. The correlation between ranks determined by students and a teacher and the correlation between the two methods of pairwise comparison were assessed using Spearman and Kendall correlation coefficients. Further, the study observed the relationship between cardinal and ordinal evaluation and the impact of reducing lambda replication. The study found that pairwise comparison is a suitable approach to evaluating peers and that it is possible to reduce the burden on evaluators using the method of incomplete comparison while maintaining a high level of correlation with the results of the complete comparison.

Keywords: cardinal and ordinal evaluation, graphical user interface design, lambda replication, pairwise comparison, peer evaluation

ŽIVOTOPIS

Laura Mandić rođena je 27. prosinca 1999. u Osijeku. Pohađala je Osnovnu školu Nikole Tesle Mirkovci, nakon koje upisuje Tehničku školu Ruđera Boškovića Vinkovci, smjer tehnička gimnazija. Tijekom osnovnoškolskog i srednjoškolskog obrazovanja sudjeluje u mnogim natjecanjima, uključujući županijska natjecanja iz matematike, informatike, engleskog jezika, hrvatskog jezika, kemije, biologije, badmintona i stolnog tenisa. Na državno natjecanje iz informatike plasira se tri puta te osvaja prvo mjesto 2016. godine, a zatim i 2018. godine. Nakon završetka srednje škole, upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tijekom studija radi kao demonstratorica na kolegijima iz područja programiranja, elektrotehnike i elektronike te kao mlađa razvojna inženjerka u privatnom sektoru. Po završetku preddiplomskog studija upisuje diplomski sveučilišni studij računarstva, izborni blok Programsko inženjerstvo. Na trećoj godini preddiplomskog studija te drugoj godini diplomskog studija od Fakulteta prima Nagradu za uspješnost u studiranju.

Laura Mandić

PRILOZI

Priloženi disk sadržava:

1. Diplomski rad „Razvoj web okruženja za ocjenjivanje i usporedbu grafičkih korisničkih sučelja“ u .docx formatu
2. Diplomski rad „Razvoj web okruženja za ocjenjivanje i usporedbu grafičkih korisničkih sučelja“ u .pdf formatu
3. Programski kod aplikacije