

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 223

**ANALIZA KORISNIČKOG ISKUSTVA I DIZAJN  
KORISNIČKOG SUČELJA U SUSTAVU EDGAR**

Ana Terović

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 223

**ANALIZA KORISNIČKOG ISKUSTVA I DIZAJN  
KORISNIČKOG SUČELJA U SUSTAVU EDGAR**

Ana Terović

Zagreb, lipanj 2021.

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 12. ožujka 2021.

**ZAVRŠNI ZADATAK br. 223**

Pristupnica: **Ana Terović (0036517166)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: izv. prof. dr. sc. Igor Mekterović

Zadatak: **Analiza korisničkog iskustva i dizajn korisničkog sučelja u sustavu Edgar**

Opis zadatka:

Napraviti pregled temeljnih načela izgradnje korisničkih sučelja posebice u kontekstu web-aplikacija. Napraviti pregled procesa razvoja korisničkog sučelja i tehnologija koje se koriste u tom procesu. Navesti dobre prakse i tipične pogreške. Objasniti odnos korisničkog sučelja (engl. user interface) i korisničkog iskustva (engl. user experience). Navesti neka od univerzalnih načela dizajna. Navesti preporuke za ostvarivanje dobrog korisničkog iskustva. Edgar je informacijski sustav koji se koristi za testiranje studenata na Fakultetu elektrotehnike i računarstva. Na temelju navedene teorijske podloge napraviti ocjenu korisničkog sučelja i doživljaja informacijskog sustava Edgar iz pozicije studenta i nastavnika. Predložiti poboljšanja i implementirati prototip poboljšanog korisničkog sučelja sustava Edgar. Donijeti ocjenu ostvarenog pristupa te smjernice za budući razvoj.

Rok za predaju rada: 11. lipnja 2021.

*Zahvaljujem se svome mentoru prof. dr. sc. Igoru Mekteroviću na savjetima i vodstvu u svakome koraku izrade ovog završnog rada. Zahvaljujem se i svima koji su sudjelovali u istraživanju te svima onima koji su mi pružili savjete.*

# SADRŽAJ

<b>1.</b>	<b>Uvod</b>	<b>1</b>
<b>2.</b>	<b>Korisničko sučelje</b>	<b>2</b>
2.1.	Proces razvoja korisničkog sučelja . . . . .	2
2.2.	Pregled tehnologija za izgradnju korisničkog sučelja web-aplikacija . . . . .	4
2.2.1.	HTML . . . . .	4
2.2.2.	CSS . . . . .	4
2.2.3.	JavaScript . . . . .	4
2.2.4.	JQuery . . . . .	5
2.2.5.	SPA (engl. <i>Single Page Applications</i> ) . . . . .	5
2.2.6.	React.js . . . . .	6
2.2.7.	Vue.js . . . . .	6
2.2.8.	Angular . . . . .	6
2.2.9.	Razvojni okviri temeljeni na CSS-u . . . . .	7
2.2.10.	Figma . . . . .	7
2.2.11.	Adobe XD . . . . .	7
2.3.	Temeljna načela izgradnje korisničkih sučelja . . . . .	8
2.3.1.	Pristupačnost (engl. <i>accessibility</i> ) . . . . .	9
2.3.2.	Dosljednost (engl. <i>consistency</i> ) . . . . .	10
2.3.3.	Vidljivost (engl. <i>visibility</i> ) . . . . .	11
2.3.4.	Prepoznavanje prije opoziva (engl. <irecognition i="" over="" recall<="">) . . . . .</irecognition>	11
2.3.5.	Smeće u smeće van (engl. <i>garbage in garbage out</i> ) . . . . .	12
2.3.6.	Progresivno otkrivanje (engl. <i>progressive disclosure</i> ) . . . . .	12
2.3.7.	Načelo trošak-dobit (engl. <i>cost-benefit</i> ) . . . . .	13
2.3.8.	Omjer signala i buke (engl. <i>signal to noise ratio</i> ) . . . . .	14
2.3.9.	Pravilo 80/20 (engl. <i>80/20 rule</i> ) . . . . .	14

2.4. Dobre prakse pri izradi korisničkih sučelja . . . . .	15
2.4.1. Poravnanje (engl. <i>alignment</i> ) . . . . .	16
2.4.2. Kompozicija (engl. <i>composition</i> ) . . . . .	17
2.4.3. Boje (engl. <i>colours</i> ) . . . . .	18
2.4.4. Naglašavanje (engl. <i>accentuation</i> ) . . . . .	19
2.4.5. Gutenbergov dijagram (engl. <i>Gutenberg diagram</i> ) . . . . .	20
2.4.6. Fittov zakon (engl. <i>Fitts' law</i> ) . . . . .	21
2.4.7. Predrasuda o obrisima (engl. <i>prejudice about outlines</i> ) . . .	22
2.4.8. Osvjetljenje odozgo-dolje (engl. <i>top-down lighting bias</i> ) . .	22
<b>3. Korisničko iskustvo</b>	<b>23</b>
3.1. Dobro korisničko iskustvo . . . . .	23
3.2. Korisničko iskustvo i korisničko sučelje . . . . .	25
3.3. Proces izgradnje dobrog korisničkog iskustva . . . . .	25
3.3.1. Definiranje zahtjeva . . . . .	25
3.3.2. Izrada alternativnih dizajna . . . . .	26
3.3.3. Izrada prototipa . . . . .	29
3.3.4. Evaluacija . . . . .	29
<b>4. Primjena na sustav Edgar</b>	<b>30</b>
4.1. Definiranje zahtjeva . . . . .	30
4.2. Izrada alternativnih dizajna . . . . .	35
4.3. Izrada prototipa . . . . .	39
4.4. Evaluacija . . . . .	40
4.5. Prijedlog za budući razvoj . . . . .	40
<b>5. Zaključak</b>	<b>41</b>
<b>Literatura</b>	<b>42</b>

# 1. Uvod

Korisnička sučelja važan su dio izgradnje web-aplikacija. Kao što im i samo ime nalaže, korisnička sučelja su namijenjena prvenstveno korisnicima. Korisnici koji se ne bave programiranjem najčešće neće razumjeti kompleksnost programa iza sučelja, ali zato će im osim funkcionalnosti biti bitan i njen izgled, preglednost i moći će cijeniti njen dizajn. Zato, bez obzira na to koliko je sustav složen, izgradnja dobrog korisničkog sučelja je od velike važnosti kako bi se u konačnici postiglo dobro korisničko iskustvo.

Cilj ovog završnog rada je napraviti sažeti pregled predloženih konvencija i praksi koje doprinose vizualnom izgledu i iskustvu uporabe korisničkih sučelja. Nakon toga, cilj je primijeniti navedene teorijske podloge na informacijskom sustavu Edgar. U tu svrhu, obrađena je literatura, provedeno je istraživanje o korisničkom iskustvu te je implementiran prototip predloženog redizajna.

Rad je podijeljen na tri dijela. Prvi dio govori o izgradnji korisničkih sučelja gdje je prikazan proces razvoja, pregled tehnologija, temeljnih načela te dobre praksi pri izradi korisničkih sučelja. Drugi dio govori o korisničkom iskustvu. Tematika korisničkog iskustva obrađena je definiranjem dobrog korisničkog iskustva, usporedbom korisničkog iskustva i sučelja te opisom procesa izgradnje dobrog korisničkog iskustva. U zadnjem dijelu je prikazana primjena prethodnih saznanja na informacijski sustav Edgar. Primjena se sastoji od provedenog istraživanja, predloženih alternativnih dizajna te implementacije prototipa.

## 2. Korisničko sučelje

Korisničko sučelje (engl. *user interface*) je skup sredstava pomoću kojih ljudi – korisnici – komuniciraju sa sustavom – određenim strojem, uređajem, računalnim programom ili drugim složenim alatom [31]. Temeljena na web-u (engl. *web-based*) korisnička sučelja ili web korisnička sučelja (engl. *web user interface*, *WUI*) su vrsta korisničkih sučelja koja prihvataju ulaz i prikazuju izlaz generiranjem web-stranica koje korisnik pregledava pomoću programa web-preglednika.

### 2.1. Proces razvoja korisničkog sučelja

Razvoj korisničkih sučelja se sastoji od dvije komponente: interakcijska komponenta i komponenta sučelja. Interakcijskom komponentom odnosno korisničkom interakcijom se definira kako funkcioniра korisničko sučelje. Korisnička interakcija se poboljšava proučavanjem smjernica, pravila, kognitivnih ograničenja i stilske interakcije. Komponenta sučelja definira način na koji su radnje korisničkog sučelja zapravo povezane s kodom koji implementira funkcije čije ponašanje korisnik promatra. Ono uključuje sprave (engl. *widget*), algoritme, knjižnice, kontrole, protok podataka, objektno orijentirani principi dizajna i uporaba jezika korisničkih sučelja [15].

Proces razvoja korisničkih sučelja se može podijeliti u tri faze:

1. projektiranje,
2. implementacija,
3. testiranje.

U fazi projektiranja se određuju funkcionalni zahtjevi aplikacije, korisnički scenariji, konceptualni, logički i fizički dizajn. Funkcionalni zahtjevi određuju inicijalne zahtjeve i ciljeve projekta, korisničkim scenarijima se predviđaju zahtjevi i kretanja korisnika a dizajnom modeliramo sustav.

U fazi implementacije se razvijaju prototipi koji se zatim konstruiraju odnosno razviju u aplikaciju. Prototipi su u početku jednostavne skice na papiru a kasnije se razvijaju u ozbiljnije i detaljnije nacrte aplikacije.

U fazi testiranja se obavlja ispitivanje upotrebljivosti (engl. *usability testing*) i ispitivanje pristupačnosti (engl. *accessibility testing*). Ispitivanjem upotrebljivosti aplikacija se testira s raznim korisnicima i scenarijima. Na primjer, testira se kupnja proizvoda u ulozi registriranog korisnika i brisanje artikla u ulozi admina. Ispitivanjem pristupačnosti provjerava se mogu li aplikaciju koristiti osobe u nepovoljnem položaju poput osoba koje ne čuju, imaju daltonizma, starije su životne dobi, itd. [12]. Obavlja se dostupnim tehnologijama i automatiziranim alatima za testiranje poput WAVE [32] i Axe [5].

## **2.2. Pregled tehnologija za izgradnju korisničkog sučelja web-aplikacija**

### **2.2.1. HTML**

HTML (engl. *Hypertext Markup Language*) je osnovni prezentacijski jezik za dokumente namijenjene za prikaz u web-pregledniku. Web-preglednici primaju HTML dokumente s web-poslužitelja ili iz lokalne pohrane i na temelju njihovog sadržaja generiraju multimedijalne web-stranice. HTML semantički opisuje strukturu web-stranice i uključuje naznake za prikazivanje dokumenta. HTML elementi kao što su `<div>`, `<span>`, `<p>` itd. su gradivni elementi HTML stranica. Parsiranjem HTML-a se izgrađuje DOM. DOM (engl. *Document Object Model*) predstavlja HTML dokument s logičkim stablom a elementi HTML-a postaju čvorovi u DOM-u [11].

### **2.2.2. CSS**

CSS (engl. *Cascading Style Sheets*) je stilski jezik koji se koristi za opis dokumenta napisanog u XML baziranom prezentacijskom jeziku kao što je HTML. CSS je dizajniran da omogući razdvajanje prezentacije i sadržaja što uključuje izgled, boje i fontove. Razdvajanje formatiranja i sadržaja omogućava predstavljanje iste HTML stranice u različitim stilovima te grupiranje istih stilova kako bi se prikazali različiti elementi [30].

### **2.2.3. JavaScript**

JavaScript (skraćeno JS) je interpretirajući, dinamički, skriptni programski jezik visoke razine često poznat kao skriptni jezik web-stranica kojima dodaje dinamička svojstva. JS je treća od standardnih web tehnologija, od kojih su prve dvije prethodno spomenuti HTML i CSS. JS omogućava stvaranje dinamički ažuriranog sadržaja, upravljanje multimedijom, animiranje slika, itd. Prilikom izgradnje korisničkog sučelja elementi web-stranice se referenciraju te im se za određeni događaj definira ponašanje dodavanjem metoda i svojstava. Sintaksa JS-a je namjerno slična i Javi i C++-u kako bi se olakšalo učenje jezika [7].

## 2.2.4. JQuery

JQuery je JS biblioteka bogatog sadržaja koja pojednostavljuje dodavanje dinamičkih svojstava web-stranicama. Biblioteka je uobičajene zadatke koji zahtijevaju mnogo redaka JS koda omotala u metode koje zahtijevaju jednu liniju koda [13]. Svoju upotrebljivost opravdava činjenicom da je trenutno koristi 77.9% web-stranica [27] no postupnim napredovanjem JS-a JQuery je u sve manjoj uporabi.

## 2.2.5. SPA (engl. *Single Page Applications*)

Višestrandične web-aplikacije (engl. *Multiple Page Applications*) rade na „tradicionalan“ način. Svaka promjena prikaza podataka ili slanja zahtjeva nazad poslužitelju zahtjeva generiranje nove stranice s poslužitelja u pregledniku što je vremenski veoma skupo. Jednostrandnične web aplikacije su nastale s ciljem ubrzavanja web-aplikacija. SPA su aplikacije koje rade unutar preglednika i ne zahtijevaju ponovno učitavanje stranice tijekom uporabe. Posjećuje se samo jedna web-stranica koja zatim učitava sav ostali sadržaj pomoću JavaScripta - o čemu uvelike ovisi. Brzina je velika prednost SPA jer se resursi učitavaju samo jedanput u životnom ciklusu aplikacije. Isto tako, lako je pronaći grešku u kodu jer je unutar web-preglednika moguće nadzirati mrežne radnje, istraživati elemente stranice i podatke povezane s njima. Još jedna prednost je da SPA može učinkovito spremiti u svoju privremenu memoriju lokalno stanje. Aplikacija šalje samo jedan zahtjev, pohranjuje sve podatke, a zatim ih može koristiti i radi čak i izvanmrežno. Neki od nedostataka SPA su incijalno sporo učitavanje zbog preuzimanja mnogo opsežnih klijentskih radnih okvira i drugih sadržaja, manja sigurnost uzrokovana XSS-om (engl. *Cross-Site Scripting*) koje omogućuje napađačima ubacivanje skripti na strani klijenta u web aplikaciju od strane drugih korisnika te curenje memorije u JS-u koje usporava sustave [18].

### **2.2.6. React.js**

React je JS biblioteka za izradu pristupnog dijela web-stranica (engl. *front-end*) odnosno korisničkog sučelja. React omogućuje manipulaciju DOM-om i upravljanje stanjima što često zahtjeva korištenje dodatnih knjižnica za ostvarivanje drugih funkcionalnosti. React se koristi za izradu jednostraničnih web-aplikacija. Glavne prednosti Reacta su njegova jednostavnost, brzina i skalabilnost. Specifičnosti Reacta su da je protok podataka jednosmjeran te da se promjena stanja komponente registrira pozivom odgovarajuće funkcije `setState()` [4].

### **2.2.7. Vue.js**

Vue je JS okvir otvorenog pristupa za izradu pristupnog dijela web-stranica odnosno korisničkog sučelja. Vue sadrži postupno prilagodljivu arhitekturu koja se fokusira na deklarativno prikazivanje i sastav komponenata. Vue dolazi s puno sintaktičkih šećera, sustavom dodataka, ugrađenim direktivama, prijelazima itd., a jezgreni tim stvorio je prateće knjižnice za usmjeravanje i upravljanje stanjima, zajedno s drugim korisnim alatima. Vue se kao i React koristi za izradu jednostraničnih web-aplikacija. Vue promjenu stanja izvršava automatski [29].

### **2.2.8. Angular**

Angular je TypeScript razvojni okvir za izradu pristupnog dijela web-stranica odnosno korisničkog sučelja. Angular je cijelovito rješenje samo po sebi kao skup TypeScript knjižnica te ne zahtijeva dodatno korištenje vanjskih knjižnica. Baziran je na MVC (engl. *Model View Controller*) modelu. Koristi se prilikom izrade jednostraničnih web-aplikacija a stanja se sama automatski mijenjanju. Razvojni okvir nije jednostavan za korištenje te nudi ograničenu količinu slobode i fleksibilnosti [2].

### **2.2.9. Razvojni okviri temeljeni na CSS-u**

Razvojni okviri za uređivanje elemenata su bazirani na CSS stilskom jeziku, te su usmjereni na izradu prilagodljivih korisničkih sučelja web-aplikacija. Sadrže čitav niz predložaka elemenata korisničkog sučelja kao što su gumbi, forme, liste, tablice i slično. Osnovna prednost njihovog dodavanja stranici je širok spektar stilsko oblikovnih mogućnosti koje olakšavaju izradu korisničkog sučelja čiji će izgled biti konzistentan neovisno o platformi i veličini zaslona na kojem se prikazuju. Najčešće korišteni razvojni okviri su Bootstrap, Materialize i Tailwind [6].

### **2.2.10. Figma**

Figma je aplikacija bazirana na pregledniku za izradu korisničkog sučelja i s njime korisničkog iskustva. Aplikacija nudi veliki broj alata za dizajn, izradu prototipova i generiranje koda. To je trenutno vodeći alat za dizajn sučelja u industriji, s robusnim značajkama koje podržavaju timove u svakoj fazi procesa dizajniranja [9].

### **2.2.11. Adobe XD**

Adobe XD odnosno Adobe Experience Design je alat namijenjen za rad na radnoj površini. Služi za dizajn korisničkog sučelja i korisničkog iskustva za web i mobilne aplikacije. Pomaže pri izradi prototipova te omogućava kolaborativni rad timova na zajedničkim projektima [1].

## **2.3. Temeljna načela izgradnje korisničkih sučelja**

Temeljna načela izgradnje korisničkih sučelja su skup uputa, savjeta i pravila koje poboljšavaju kvalitetu korisničkih sučelja. Navedena načela i njihov redoslijed su temeljeni na radovima poznatih znanstvenika u području izgradnje korisničkih sučelja, Jakobu Nielsenu i Benu Shneidermanu. Jakob Nielsen napisao je *10 heuristika upotrebljivosti za dizajn korisničkog sučelja* [20] a Ben Shneiderman je napisao *8 zlatnih pravila dizajna korisničkih sučelja* [25]. Za odabir načela je isto tako od velike pomoći bila i knjiga *Univerzalna načela dizajna* autora Williama Lidwell, Kritine Holden i Jill Butler [14].

Prilikom objašnjavanja svakog načela naveden je i primjer dobre ili loše uporabe tog načela u korisničkom sučelju sustava Edgar.

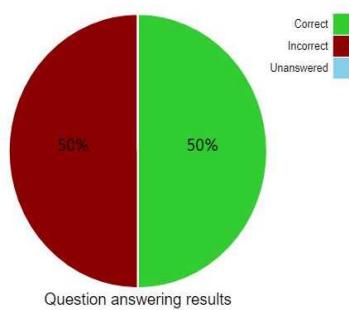
### 2.3.1. Pristupačnost (engl. *accessibility*)

Načelo pristupačnosti nalaže da dizajn treba biti uporabljiv osobama različitih sposobnosti. Pristupačan dizajn svojom fleksibilnošću treba biti uporabljiv i novom i stručnom korisniku, različitim dobnim skupinama, osobama s invaliditetom, stranim korisnicima i tehnološki različito educiranim korisnicima. Prema Europskom parlamentu [24] postoje četiri načela pristupačnosti web-stranica:

1. mogućnost percepcije,
2. operabilnost,
3. razumljivost,
4. stabilnost.

Mogućnost percepcije se osigurava prikazom sadržaja koji je jasan i korisnicima sa samo jednim osjetilom za percepciju, operabilnost dodavanjem više alata za upravljanje sadržajem, razumljivost kada se koristi jezik koji je jasan i dosljedan sučelju, a stabilnost se postiže kada je web-stranica upotrebljiva na različitim platformama.

Upotreba boja na stranici učestali je problem za pristupačnost. Svaki čovjek ima drugačiju percepciju boje, neki ne mogu vidjeti cijeli spektar boja a neki ljudi uopće ne mogu vidjeti boje. Zato je prilikom prikaza sadržaja važno ne koristiti samo boje kako bi prikazali informacije. Korisnik koji ne raspoznaće dobro boje na slici 2.1 ne može iščitati informacije o kojim podacima se radi u grafikonu.



**Slika 2.1:** Graf udjela riješenosti zadatka u Edgaru

### 2.3.2. Dosljednost (engl. consistency)

Načelo dosljednosti govori da u sličnim situacijama se trebaju izvršiti iste radnje kako korisnici ne bi trebali razmišljati znače li različite radnje ili riječi istu stvar. Različiti elementi međusobno trebaju biti primjetljivo različiti a elementi iste funkcije trebaju dijeliti vizualni identitet.

Razlikuje se vanjska i unutarnja dosljednost. Vanjska dosljednost se poboljšava koristeći konvencije uspostavljene zajednicom, industrijom i platformom. Jakobov zakon govori kako ljudi provode najviše svoga vremena koristeći digitalni proizvod koji nije naš [19]. To znači da su očekivanja korisnika kako treba koristiti proizvod bazirana na iskustvu ostvarenom kod nekoga drugoga. Unutarnja dosljednost se poboljšava konzistentnim korištenjem elemenata kroz cijeli proizvod. Dosljednost se odnosi na terminologiju, boje, raspored, fontove, itd. [21].

U sustavu Edgar gumb za brisanje je primjer i vanjske i unutarnje nedosljednosti. Problematika vanjske dosljednosti je u tome da se gumb za brisanje najčešće nalazi kao zadnji na listi mogućnosti dok se u sustavu Edgar tamo ne nalazi. Unutarnja dosljednost je u ovome primjeru narušena nekonzistentnim pozicioniranjem gumba za brisanje u različitim komponentama.

#	Id	Title	Created	Test type	Test ordinal	Abbrev.	Max runs (0=–)	Global Public UseStats Scoreignored	Show solutions	Max score	Password	No. of qs	Duration (s)	Valid from	Valid to	Delete	Edit
1	13162	Hack It test za isprobavanje	Igor Mekterović	Evaluated C-lang questions	1	DBG	100	-- STATS SCIGNORED		4,00	123***	4	60000	2021-02-06 11:50:20	2022-02-06 11:50:20	<button>Delete</button>	<button>Edit</button>
2	13255	Evin test	Ana Terović (student)	Evaluated Java questions	2	ET	3	GLBL- STATS SCIGNORED		20,00	edg***	2	600	2021-05-24 22:00:00	2021-05-25 23:59:00	<button>Delete</button>	<button>Edit</button>
3	13257	Generated, please update...	Ana Terovic (student)	Classic ABC questions	3		1	-- STATS-		100,00	kQK***	10	600	2021-05-25 03:31:24,046596	2021-05-25 03:41:24,046596	<button>Delete</button>	<button>Edit</button>

Slika 2.2: Tablica definiranih ispita u Edgaru

#	Id	Ordinal	Title	Type	Score ignored	Total	Started %	Submitted %	Delete All	Forward generate	Generated %	Files uploaded	Submitted %	Ongoing %	Not started %	Send message	Manual grading	Tickets %
1	13257	3	Generated, please update...	Classic ABC questions	false	4	0.00	0.00	<button>Delete All</button>	<button>Forward generate</button>	0	0	0	0	4			
2	13255	2	Evin test	Evaluated Java questions	true	4	50,00	50,00	<button>Delete All</button>	<button>Forward generate</button>	2	0	2	0	2			
3	13162	1	Hack It test za isprobavanje	Evaluated C-lang questions	true	4	25,00	0,00	<button>Delete All</button>	<button>Forward generate</button>	1	0	0	1	3			

Slika 2.3: Tablica instanci ispita u Edgaru

### 2.3.3. Vidljivost (engl. *visibility*)

Upotrebljivost sustava poboljšava se kad su njegov status i načini upotrebe jasno vidljivi. Korisnici bi u svakome trenutku trebali znati gdje se nalaze i što se događa kroz prikladno sučelje i povratne informacije. Poznavanjem trenutnog statusa sustava rezultira korisnikovo mogućnosti da predvidi što sve treba ubuduće napraviti kako bi postigao željeni rezultat. Predvidivost sustava stvara korisnikovo povjerenje u proizvod.

Na slici 2.4 je prikazana statusna traka prilikom rješavanja ispita u sustavu Edgar što je primjer dobre primjene načela vidljivosti.



**Slika 2.4:** Statusna traka prilikom rješavanja ispita u sustavu Edgar

### 2.3.4. Prepoznavanje prije opoziva (engl. )

Načelo prepoznavanje prije opoziva govori da korisnici elemente sučelja trebaju prepoznati a ne prisjećati ih se. Prepoznavanje znači da se može „prepoznati“ da je nešto poznato dok prisjećanje zahtjeva „dobavljanje“ informacija iz memorije. Prisjećanje je zato kognitivno zahtjevnije te time otežava uporabu korisničkih sučelja.

Slika 2.5 je primjer narušenog načela u sustavu Edgar. Popis svih pitanja naorušava načelo prepoznavanja jer korisnici neće moći prepoznati o kojem je pitanju riječ kada vide samo ID pitanja već će se morati prisjetiti koji ID pitanja je vezan uz koje pitanje.

### Question Ids (cnt = 19). Hover the ID for more info.

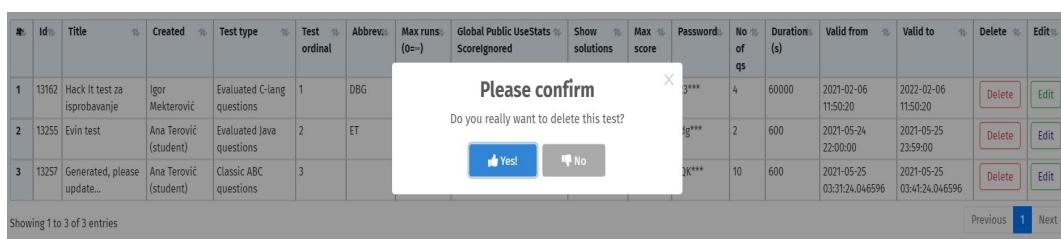
42896 42920 42927 42946 42947 42956 43884 43985 43985 43988 43991 44231 44233 44233 44233 44234 44234 44234 44235

**Slika 2.5:** Popis svih pitanja u Edgaru

### 2.3.5. Smeće u smeće van (engl. *garbage in garbage out*)

Smeće u smeće van je fraza koja govori da kvaliteta unosa u sustav određuje kvalitetu proizvoda sustava. Kontrola unosa se realizira kvalitetnim povratnim informacijama u slučaju napravljenе pogreške te mogućnosti povratka u slučaju da korisnik shvati da to nije akcija koju je htio napraviti. Zaštitom korisnika od pogrešaka mu se daje osjećaj sigurnosti i slobode prilikom korištenja sučelja.

Primjer dobro korištenog načela smeće u smeće van prikazuje slika 2.6. U sustavu Edgar prilikom obavljanja akcija kao što su brisanje elemenata pojavljuje se provjeravajuća poruka u slučaju da korisnik to ne želi napraviti.



Slika 2.6: Brisanje elementa u Edgaru

### 2.3.6. Progresivno otkrivanje (engl. *progressive disclosure*)

Progresivno otkrivanje znači da elementi unutar korisničkog sučelja trebaju napredovati prirodno, od jednostavnog prema složenom jer to oponaša prirodan način na koji mozak sukcesivno obrađuje informacije. U bilo kojem trenutku prikazuju se samo potrebne ili nužne informacije.

Primjer narušenog načela progresivnog otkrivanja u sustavu Edgar se vidi na slici 2.7. Prilikom odabira „Exam definition“ u navigaciji otvara nam se cijeli niz mogućnosti koje je moguće raspodijeliti u jednostavnije zasebne dijelove.

Slika 2.7: Forma za izradu ispita

### 2.3.7. Načelo trošak-dobit (engl. *cost-benefit*)

Aktivnost će biti izvršena samo ako je dobit jednaka ili veća od troška. Pri-likom izvršavanja neke radnje korisnik ne bi trebao raditi ništa više od onoga što je nužno. U suprotnom se korisnik obeshrabruje i stranica mu pruža preve-liki napor. Povećanje dobiti se može ostvariti povratnim informacijama prilikom završetka ispunjavanja neke akcije. One pružaju korisniku zadovoljstvo postignu-ćem i osjećaj olakšanja. Primjer načela je pravilo tri klika (engl. *three click rule*) koje govori da ako korisnik ne može pronaći željenu informaciju kroz tri klika postat će frustriran te će napustiti stranicu.

Primjer narušenog načela trošak-dobit u sustavu Edgar prikazuje slika 2.8. Forma za dodavanje studenta na predmet zahtjeva veliki kognitivni napor te unos velikog broja informacija.

```
1 # Provide a list of students with the following columns:
2 # ALT_ID    alt_id2 FIRST NAME LAST NAME   email   group_name auth_provider
3 # eg
4 # username@domain  0012345678  Edgar   Codd   edgar@mail.com 601 google
5 # someotheruser@domain      John   Doe
6
7 # Note that:
8 #   * alt_id is actually USERNAME, obviously depends on what auth provider are you using
9 #   * alt_id2 is typically another identificator used by the higher education institution, some kind of academic id
10 #   * alt_id, and alt_id2 if provided, MUST BE UNIQUE
11 #   * auth_provider is authentication provider, eg saml.aai, google, facebook, twitter, ...
12 #   * uppercase columns are obligatory,
13 #   * fields are tab-separated,
14 #   * rows are (CR)LF separated
15 #   * comments start with #, so you can leave or delete these instructions"
```

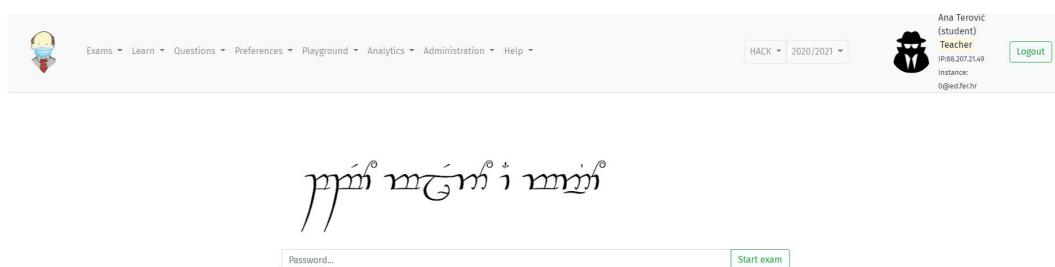
Parse

Slika 2.8: Forma za dodavanje studenata na Edgaru

### 2.3.8. Omjer signala i buke (engl. *signal to noise ratio*)

Cilj dobrog dizajna jest postići veliku vrijednost omjera signala naspram buke maksimiziranjem signala te minimiziranjem buke. Minimiziranje buke znači uklanjanje nepotrebnih elemenata i maksimiziranje izražaja nužnih elemenata. Svaki nepotrebni dodatak, slika, linija ili simbol, odvraća pažnju od važnih elemenata. Horror-vacui (lat. *strah od praznog prostora*) koji se definira kao sklonost ispunjavanju praznog prostora objektima i elementima umjesto da se prostori ostave praznima, je primjer negativnog utjecaja na omjer signala i buke. Što horror vacui više raste, to se opažana vrijednost smanjuje.

Primjer dobro korištenog načela omjera signala i buke u sustavu Edgar se vidi na slici 2.9 . Prilikom odabira opcije za započinjanje ispita otvara se sučelje koje ima samo jednu funkciju te se pažnja korisnika ne odvraća nepotrebnim sadržajima.



Slika 2.9: Forma za početak ispita u Edgaru

### 2.3.9. Pravilo 80/20 (engl. *80/20 rule*)

Pravilo 80/20 govori da je veliki postotak efekata uzrokovan malim postotkom varijabli. Konkretno, 80 posto korištenja proizvoda uključuje 20 posto njegovih značajki [14]. Načelo je izrazito korisno prilikom izrade proizvoda gdje su resursi limitirani jer omogućuje raspoznavanje elemenata na koje se treba fokusirati te na koje trebamo potrošiti resurse kako bi dobili najveći pozitivni efekt.

Primjer narušenog pravila 80/20 u sustavu Edgar prikazan je na slici 2.7. Kroz elemente nije uspostavljena hijerarhija važnosti i nisu zamjetno kategorizirani pa je zbog toga teško pronaći željenu kontrolu ili element.

## 2.4. Dobre prakse pri izradi korisničkih sučelja

Dobre prakse su praktične i konkretnе preporuke za izradu korisničkih sučelja. Lista se temelji na geštalt načelima percepcije te knjizi *Univerzalna načela dizajna* [14]. Psihologija geštalta govori da je cjelina važnija od dijela (engl. *the whole is more than the sum of its parts*). Osnovna načela geštalt percepcije su:

1. sličnost (engl. *similarity*),
2. kontinuitet (engl. *continuation*),
3. zatvorenost (engl. *closure*),
4. bliskost (engl. *proximity*),
5. zajedničko kretanje (engl. *common fate*),
6. figura i pozadina (engl. *figure/ground*).

Primjenom načela geštalta se dizajn koji se čini nasumičnim ili kao da se bori za pažnju korisnika može promijeniti u onaj koji nudi neprimjetnu, prirodnu interakciju zbog koje će web-stranica imati poznati osjećaj, usmjeravajući korisnike prema radnji koja se od njih očekuje [10].

### 2.4.1. Poravnanje (engl. *alignment*)

Uspostavom zajedničkog poravnajanja elemenata stvara se dojam njihovog međusobnog zajedništva. Mogu se izdvojiti dvije vrste poravnajanja: (1) poravnanje po stupcima i linija i (2) poravnanje po području. Poravnanje prema stupcima i linijama obuhvaća horizontalno, vertikalno i poravnanje po središtu. Poravnanje prema ukupnomu području se koristi kod elemenata koji su asimetrični i nejednolici. Element se treba pozicionirati s obzirom na njegovu os ravnoteže što znači da je sa svake strane vizualno jednaka težina.

Dobar primjer primjene poravnajanja u sustavu Edgar se vidi prilikom odabira boje koda prikazano na slici 2.10.

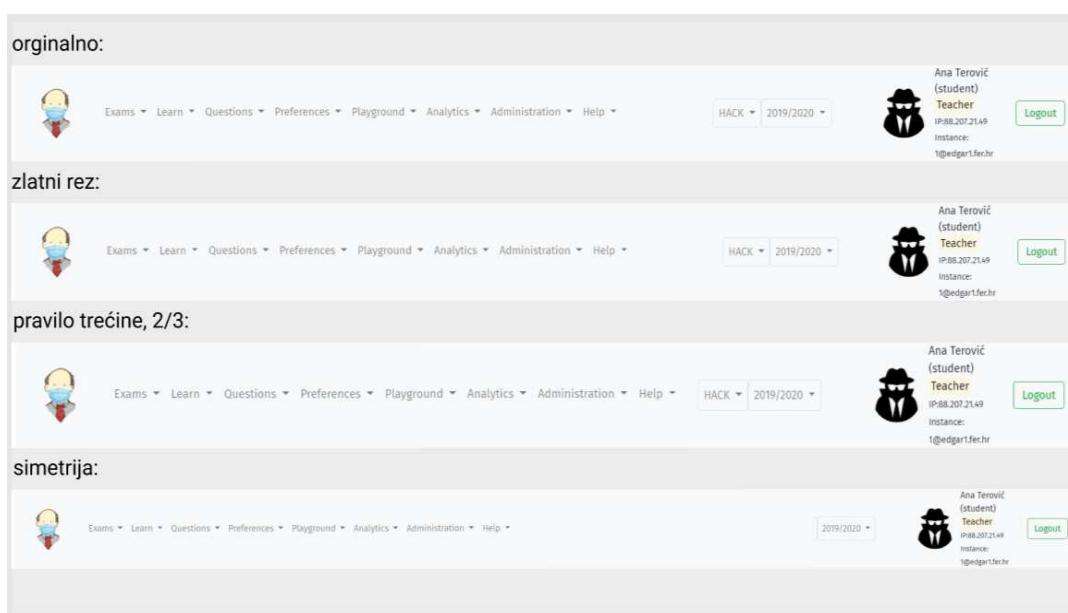
<input type="radio"/> 3024-day	<input type="radio"/> 3024-night	<input type="radio"/> abcdef	<input type="radio"/> all-hallow-eve
<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>
<input type="radio"/> ambiance-mobile	<input checked="" type="radio"/> ambiance	<input type="radio"/> amy	<input type="radio"/> argonaut
<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>
<input type="radio"/> arona	<input type="radio"/> base16-dark	<input type="radio"/> base16-light	<input type="radio"/> bbredit
<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>	<pre>1 SELECT *, 'string' 2 FROM student 3 WHERE id = 123; 4 -- comment</pre>

Slika 2.10: Primjer poravnajanja u sustavu Edgar

## 2.4.2. Kompozicija (engl. *composition*)

Kompozicija je suradnja dijelova s ciljem kako bi se prenijelo značenje. Istaknuli su se tri tipa kompozicije: (1) zlatni rez, (2) pravilo trećine i (3) simetrija. Zlatni rez predstavljamo omjerom među elementima u iznosu od otprilike 0,618. Pravilo trećine, kao što i samo ime govori, dijeli element na trećina čime stvara estetski privlačne položaje za primarne elemente dizajna. Simetrija je najosnovniji aspekt univerzalne ljepote. Prikladnom primjernom simetrije stvara se ravnoteža, harmonija i stabilnost.

Prethodno navedene kompozicije su primijenjene na navigaciju Edgar. Na slici 2.11 se vidi kako originalno sučelje zapravo prati pravilo zlatnog reza te je primjer dobrog omjera bijelog prostora i sadržaja. S druge strane pravilo trećine ne ostavlja dovoljno bijelog prostora između segmenata dok primjenom simetrije imamo previše bijelog prostora.

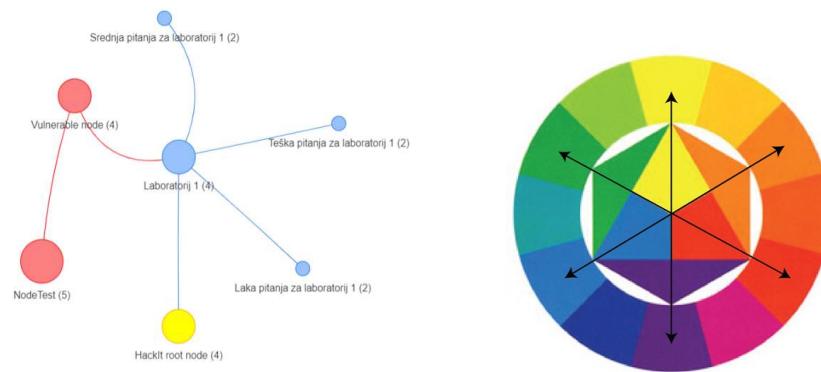


Slika 2.11: Primjer kompozicija u sustavu Edgar

### 2.4.3. Boje (engl. *colours*)

Pravila kombiniranja boja biti će prikazana uz pomoć kotača boja. Kotač boja je apstraktna ilustrativna organizacija nijansi boja oko kruga, koja pokazuje odnose između njih. Estetski privlačne kombinacije su susjedne boje, oprečne odnosno komplementarne boje, boje na kutovima simetričnog mnogokuta odnosno trokut i četverokut te prirodne kombinacije. U kontekstu topline boje, treba koristiti toplije boje za elemente u prednjem planu a hladnije boje za elemente u pozadini. Zasićenije boje izražavaju uzbudljivost, dinamičnost te traže pažnju dok nezasićene boje iskazuju rad, učinkovitost, profesionalizam i ugodu.

Dobar primjer korištenja komplementarnih boja u sustavu Edgar se vidi na slici 2.12 gdje su za grafički prikaz Node-ova korištene primarne boje crvena, plava i žuta koje u kotaču boja čine simetrični trokut.



**Slika 2.12:** Primjer dobre primjene boja u sustavu Edgar

## 2.4.4. Naglašavanje (engl. *accentuation*)

Naglašavanje je tehnika svraćanja pozornosti na područje teksta ili slike. Naglašava se 10 posto vidljivog dizajna a što se više toga ističe time se više gubi efekt. Tehnike naglašavanja su:

1. podebljanje, kurziv, podcrtavanje;
2. font;
3. boja;
4. treperenje.

Važno je napomenuti kako treba izbjegavati korištenje različitih fontova kroz isto sučelje a stavljanje elemenata u negativ dobro funkcionira s tekstom, ali ne tako dobro s ikonicama ili oblicima.

Slika 2.13 prikazuje dobru primjenu naglašavanja u sustavu Edgar. Pri otvaranju Help/Glossary kartice prikazuje se velika količina teksta gdje je podebljanjem određenih riječi korisniku olakšano čitanje.

### Glossary

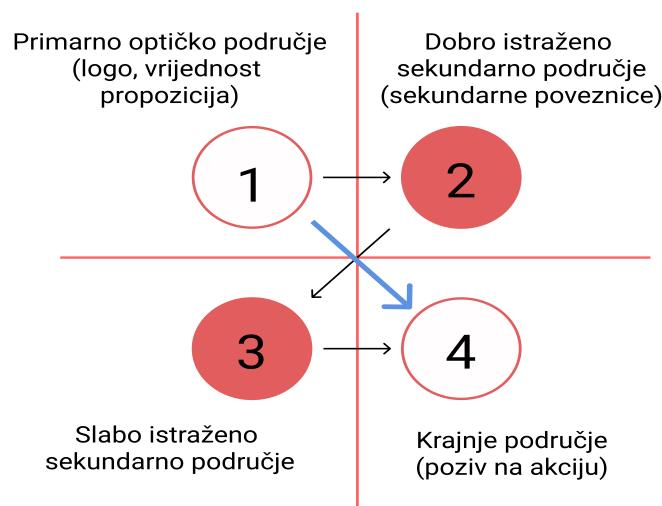
- Exam
  - **Exam (definition)** is a definition of parameters (password, exam length, no of questions, question pool) needed to produce a exam for a student. (analogy - Class)
  - Exam consists of a number (1..N) of **exam parts**. **Exam part** defines a question node that will be used as a pool of random questions. Exam part also defines a min/max number of questions drawn from the pool, pass percentage and grading model. Exam part has "ordinal" field which is by default set to 0 for all test parts. This attribute defines the ordering of generated questions. If left default, question will be randomly ordered. Otherwise, questions will be ordered by the test part ordinal and randomly within the test part. In SQL dialect, one might say that generated questions will be ordered "BY test\_part.ordinal, random()".
  - **Exam instance** is a 'materialization' of an exam, always produced for a single student. Exam materialization usually involves random question selection (from a predefined pool). (analogy - Object)
  - **Grading model** defines (correct points, incorrect points, unanswered points) triplets. Note that a single exam can have multiple exam parts having different models. One must take care that the cumulative number of points is correct and deterministic.
- Question
  - **Question** is a definition of a problem assignment, with the (hidden) correct answer. Currently, multiple-choice questions and SQL question are supported. Questions are written in GitHub markdown syntax (which supports tables). Additionally, Edgar supports images and inline TEX formulas via MathJax library (formulas must be enclosed in \$ tags).
  - Question belongs to a number of **question nodes**. **Question nodes** can have a number of parents and children, that is - they form a network (graph), and can be of arbitrary types (eg course, module, unit, ...).
  - **Question nodes** are used to categorize questions in a manageable "bins" or "pools"; see Exam parts above.
- Tutorial
  - **Tutorial** is a definition of its title, description and steps that need to be completed in the given order by students to finish the tutorial.
  - A tutorial is assigned to the course where it is defined but can be assigned to multiple courses (not yet using the Web UI).
  - An active tutorial is listed to students in the assigned course(s) while inactive tutorials are not listed.
  - **Steps** are defined by their title and text and usually explain a concept from the course material the tutorial is about. Every step contains one question (equal to the above described concept) that students need to solve correctly in order to proceed to the next tutorial step. Steps and questions contain several types of hint that can be active (shown to students when appropriate) or inactive (ignored).
  - **Step hints** are defined as guidelines for students on how to solve the given question. A step can contain several step hints and they are shown to students on their request in the defined order.
  - **Question hints** serve the same purpose, but are shown depending on student-supplied answer to the question. There are two types of step hints:
    - **Question answer hints** used in questions of (single or multiple choice) ABC question type. For each possible answer, a hint can be defined and shown if the student chose that answer.
    - **Code question answer hints** used in code question types (SQL, C-Lang, Java...). The hints are shown if the given regular expression expression matches the student-supplied answer.

**Slika 2.13:** Primjer naglašavanja u sustavu Edgar

## 2.4.5. Gutenbergov dijagram (engl. *Gutenberg diagram*)

Gutenbergov dijagram prikazan slikom 2.14 opisuje glavni smjer kretanja pogleda prilikom gledanja ravnomjerno raspoređenih, homogenih informacija. Dijagram dijeli vidno područje na 4 kvadranta: glavno optičko područje u gornjem lijevom kutu, završno područje u donjem desnom kutu, snažno slijedno područje u gornjem desnom kutu i slabo slijedno područje u donjem lijevom kutu. Ovo načelo nam služi kao pomoć prilikom kompozicije elemenata koji su ravnomjerni a u ostalim slučajevima se treba usmjeravati prema veličini i kompoziciji elemenata.

Slika 2.15 prikazuje formu za izradu ispita što je primjer koji narušava Gutenbergov dijagram jer se gumbi za akciju nalaze u gornjem desnom kutu dok je preporučena pozicija u donjem desnom kutu.



Slika 2.14: Gutenbergov dijagram

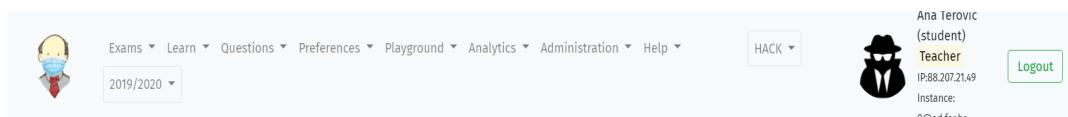
Exam definition												Clone	Clone up	Validate	Save	Test Generate		
Id: <b>13248</b> Created: Ana Terović (student) (2021-05-20 12:13:15) Type: Classic ABC questions      Modified: edgar (2021-05-20 12:16:26)																		
Ordinal	Title	Title abbrev				Password				Available from		Available to						
2	Demo									R7XvPwI4wLXXJhp86mFbzj8I		2021-05-20 12:13:15		2021-05-30 12:23:15				
Global	Public	Use in stats	Show solutions	Hint result	Async submit	Score ignored	Forward only	Anon. stalk	Question pace	Test pace								
<input type="checkbox"/> Of	<input type="checkbox"/> Of	<input checked="" type="checkbox"/> On	<input type="checkbox"/> Of	<input checked="" type="checkbox"/> On	<input checked="" type="checkbox"/> On	<input type="checkbox"/> Of	<input type="checkbox"/> Of	<input type="checkbox"/> Of	<input type="checkbox"/> None	<input type="checkbox"/> None								
Max runs	Max score	No of questions	Duration (secs)	Pass perc	Rvw period (min)	Upload files	Upload limit(B)	Email reminder	Trim clock									
1	1.00	1	600	0.0	10	0	2097152	None	<input type="checkbox"/> Of									

Slika 2.15: Primjer narušenog Gutenbergovog dijagraama u sustavu Edgar

## 2.4.6. Fittov zakon (engl. *Fitts' law*)

Fittov zakon govori kako vrijeme potrebno da se stigne do cilja jest funkcija veličine mete i udaljenosti do mete. Postavljanje gumba uz rub ili u kut zaslona znatno smanjuje količinu potrebnog kretanja što rezultira rjeđim pogreškama i češćim postizanjem cilja. S obzirom na upotrebu, češće korištene kontrole trebaju biti blizu i dovoljno velike a obratno za rjeđe korištene kontrole.

Slika 2.16 prikazuje dobру primjenu Fittovog zakona. Gumb za odjavu iz sustava se nalazi na prikladnoj poziciji uz rub ekrana te je prikladne veličine.



**Slika 2.16:** Primjer primjene Fittovog zakona u sustavu Edgar

#### 2.4.7. Predrasuda o obrisima (engl. *prejudice about outlines*)

Predrasuda o obrisima je sklonost davanja prednosti objektima s mekom linijom obrisa umjesto onima s oštrim kutevima ili vrhovima. Oštri rubovi traže pažnju i potiču na razmišljanje dok meki rubovi ostavljaju pozitivan prvi dojam.

#### 2.4.8. Osvjetljenje odozgo-dolje (engl. *top-down lighting bias*)

Osvjetljenje odozgo-dolje označava sklonost tumačenja osjenčanih ili tamnih područja nekog predmeta kao sjena koje su posljedica izvora svjetlosti iznad tog predmeta. Koristi se jedan izvor svjetlosti u gornjem lijevom kutu za prirodni i funkcionalni predmeti ili okoline. Donji izvori svjetlosti se koriste za prikaz neprirodnih ili prijetećih predmeta ili okoline.

Korisničko sučelje Edgar nema primjer ovoga načela ali slika 2.17 prikazuje kako osvjetljenje odzogo-dolje utječe na slike koje su često dio web-stranica.



Slika 2.17: Primjer osvjetljenja odozgo-dolje, slika preuzeta od [26]

# 3. Korisničko iskustvo

Prema izumitelju samoga naziva, Donu Normanu, korisničko iskustvo (engl. *user experience*) se definira kao svi aspekti interakcije krajnjeg korisnika s tvrtkom, njezinim uslugama i proizvodima [23]. Inicijativa za proučavanje korisničkog iskustva nastala je kada su u Appleu shvatili da je iskustvo korištenja kompjutera bilo vrlo loše: kompjuteri su bili glomazni i izgledali su zastrašujuće, od trenutka kada su korisnici glomazni kompjuter nekako trebali staviti u auto, do otvaranja kutije i suočavanja s njegovim zastrašujućim izgledom sve do skupljanja hrabrosti kako bi ga sastavili i napokon počeli koristiti. Sve je to bio dio lošeg korisničkog iskustva zbog kojeg je naposljetku po prvi put osnovan odsjek posvećen korisničkom iskustvu predvođen prethodno spomenutim, Donom Normanom [22].

## 3.1. Dobro korisničko iskustvo

Cilj korisničkog iskustva je osigurati da korisnici pronađu osobnu vrijednost u proizvodu koji im se pruža. Proizvod visoke vrijednosti ispunjava daleko više od samo onoga što korisnik zahtjeva [23].

Faktore koji utječu na vrijednost korisničkog iskustva predstavio je Peter Morville u svojim *Saćama korisničkog iskustva* (engl. *User Experience Honeycomb*) [16]. Sače se sastoje od sedam célija:

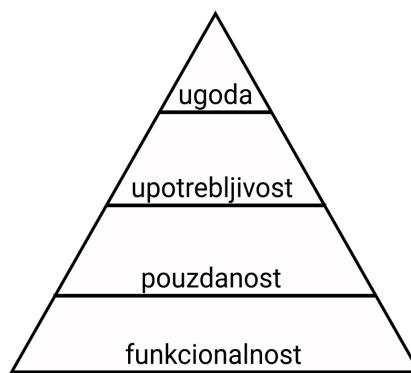
1. korisnost (engl. *useful*),
2. upotrebljivost (engl. *usable*),
3. poželjnost (engl. *desirable*),
4. pronalažljivost (engl. *findable*),
5. pristupačnost (engl. *accessible*),
6. vjerodostojnjost (engl. *credible*)

koji okružuju centralnu vrijednosnost (engl. *valuable*). Korisni proizvod ispunjava neku svrhu, iskoristivi se lako koristi, poželjni je vizualno privlačan i primjetan, pronalažljiv je jednostavan za navigaciju, pristupačan je dostupan svim ljudima, a vjerodostojan je onaj u koji korisnik vjeruje i ima povjerenje.



**Slika 3.1:** Sače korisničkog iskustva

Alternativno objašnjenje dobrog korisničkog iskustva predstavio je i Aarron Walter u svojoj knjizi *Dizajniranje za osjećaje* gdje je napravio varijaciju Maslowe hijerarhije potreba. Na dnu njegove piramide korisničkih potreba se nalazi funkcionalan (engl. *functional*) proizvod, karakteristika sljedećeg sloja je pouzdanost (engl. *reliable*), zatim upotrebljivost (engl. *usable*) te na vrhu piramide je ugodan (engl. *pleasurable*) proizvod [8].



**Slika 3.2:** Aaron Walterova hijerarhija korisničkih potreba

Uz navedene faktore, da bi se postiglo visokokvalitetno korisničko iskustvo, mora postojati besprijekorno spajanje usluga više disciplina, uključujući inženjering, marketing, grafički i industrijski dizajn i dizajn sučelja [23].

## **3.2. Korisničko iskustvo i korisničko sučelje**

Pojmovi korisničko iskustvo (engl. kratica *UX*) i korisničko sučelje (engl. kratica *UI*) se često pogrešno koriste naizmjenično kao sinonimi. Korisničko iskustvo se odnosi na sveukupno iskustvo tijekom korištenja nekog proizvoda dok se korisničko sučelje odnosi na trenutak interakcije čovjeka sa strojem te vizualan prikaz te interakcije. Proces izgradnje dobrog korisničkog iskustva obuhvaća izgradnju korisničkog sučelja, a kako bi se ostvarilo dobro korisničko sučelje potrebne su informacije dobivene prethodnim istraživanjima o korisničkom iskustvu. Važno je razlikovati ukupno korisničko iskustvo od korisničkog sučelja, iako je korisničko sučelje očito izuzetno važan dio sveukupnog dojma proizvoda.

## **3.3. Proces izgradnje dobrog korisničkog iskustva**

Proces izgradnje dobrog korisničkog iskustva temelji se na uzastopnom iteriranju 4 faze:

1. definiranje zahtjeva,
2. izrada alternativnih dizajna,
3. izrada prototipa,
4. evaluacija.

### **3.3.1. Definiranje zahtjeva**

Cilj definiranja zahtjeva je dobiti što bolje razumijevanje korisnika, njegovih zah-tjeva i trenutnih praksi. Na ovaj se korak također može gledati kao na razumi-jevanje problemskog prostora – što koči dovršenje zadatka, kako se zadatak ili proces može poboljšati. Prikupljamo podatke kako bismo razumjeli sva moguća mesta na kojima možemo poboljšati korisničko iskustvo.

Razumijevanje korisnika ostvarujemo prikupljanjem informacija kroz razne tehnike istraživanja kao što su intervju, fokus grupe, upitnici i promatranje u prirodnom okruženju. Intervju s korisnikom omogućava jedan na jedan diskusiju

te se zato provodi kada se žele prikupiti detaljne informacije o njegovu iskustvu. Fokus grupe se sastoje od pet do deset korisnika koje se zatim usmjerava u međusobnu diskusiju. Među najvećim prednostima fokus grupe je to što dizajner može prikupiti mnogo podatka o temi koja ga zanima u relativno kratkom vremenskom razdoblju. Također, grupna dinamika može rezultirati podacima koji možda ne bi bili prikupljeni u pojedinačnom intervjuu. Upitnici se provode kako bi se od korisnika dobio izvještaj o vlastitom ponašanju, percepciji, stavovima i osjećajima. Upitnici omogućavaju da se istovremeno prikupe podatci od određenog broja pojedinaca te je analiza podataka prilično brza. Cilj naturalističkog promatranja je promatranje korisnika u akciji u njegovom vlastitom okruženju. Umjesto da pita korisnika što, kada i gdje izvršava zadati zadatak, dizajner izlazi tamo gdje se zadatak odvija i promatra korisnika. Naturalističko promatranje eliminira mogućnost utjecaja ispitanika na mišljenje korisnika. Nakon prikupljanja, podatke je potrebno analizirati i obraditi tehnikama kao što su deskriptivna statistika, korisničke tablice i persona. Korisničke tablice sadrže popis svih korisnika i informacija o njima. Persone su arhetipski korisnici čiji ciljevi i karakteristike predstavljaju potrebe veće skupine korisnika. Obradeni podatci se mogu prezentirati u obliku scenarija, scenarija bitnih slučajeva upotrebe, hijerarhijske analize zadataka ili trenutnom kritikom korisničkog sučelja.

### 3.3.2. Izrada alternativnih dizajna

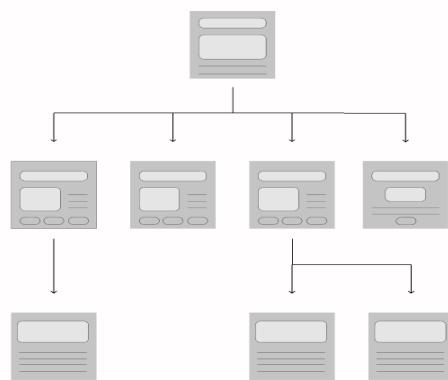
Cilj faze alternativnog dizajna je razviti sučelja ili sustave koji će bolje ispuniti potrebe korisnika od onih koji već postoje. Primarni zadatak dizajniranja alternativa je koncipiranje osnove dizajna. Ovdje se pregledavaju podatci iz prve faze i utvrđuje se što korisnik nalaže da mu treba. Podaci iz prve faze također omogućuju utvrđivanje funkcionalnih i nefunkcionalnih zahtjeva za dizajnerske alternative. Funkcionalni zahtjevi su stvari koje bi sustav trebao raditi. Oni se temelje na očekivanom funkcioniranju sustava. Nefunkcionalni zahtjevi se odnose na ograničenja na sustav i njegov razvoj.

Jedna od disciplina koja se koristi prilikom dizajna alternativa je arhitektura informacija. Arhitektura informacija se definira kao strategija za svjesno organiziranje sadržaja i protoka web-stranice, aplikacije ili sustava [28]. Louis Rosenfeld i Peter Morville su arhitekturu informacija podijelio na četiri komponente [17]:

1. organizacijske strukture koje definiraju kako će podaci biti organizirani,
2. označavanje sustava koji definiraju način prikaza informacija u sustavu,

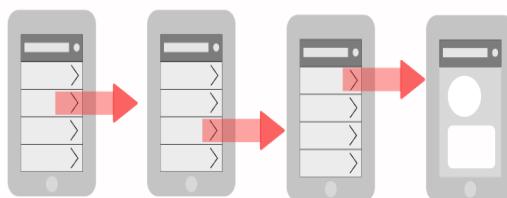
3. navigacijski sustavi koji definiraju kako će se korisnici kretati kroz informacije,
4. pretražujući sustavi koji definiraju kako će korisnik pronaći podatke.

Postoji nekoliko glavnih modela organizacijskih struktura: hijerarhijska, sekvencijska i matrična. Hijerarhijska struktura prikazana na slici 3.3 se temelji na postavljanju elemenata na takav način da korisnici mogu razumjeti njegovu razinu važnosti. U primjeni se koristi kada web-stranice sadrže mnogo podataka te postoji mnogo različitih puteva.

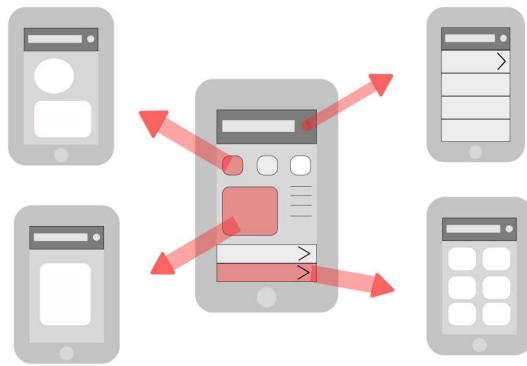


**Slika 3.3:** Hijerarhijska struktura

Sekvencijska struktura prikazana na slici 3.4 stvara određenu vrstu puta za korisnike. Korisnici prolaze korak po korak kroz sadržaj kako bi izvršili zadatku koji im je potreban. U primjeni se često koristi na stranicama za kupovinu kako bi se izvršila kupnja nečega.



**Slika 3.4:** Sekvencijska struktura



**Slika 3.5:** Matrična struktura

Matrična struktura prikazana na slici 3.5 omogućava korisniku da sam odabire svoj put kroz različiti izbor sadržaja. U primjeni se koristi kada je potreban jedan visoko namjenski zaslon. U praksi je najčešće kombiniranje različitih modela, ali treba pripaziti kako se ujedno i korisnika ne bi zbunilo.

Cilj sustava za označavanje je učinkovito objediniti podatke. Istovremeni prikaz mnogo informacija zbunjuje korisnika te se zbog toga stvaraju oznake koje u nekoliko riječi mogu predstaviti puno podataka. Primjer je oznaka Kontakt u navigacijama koja za korisnika označava cijeli skup informacija kao što je email, broj mobitela, adresa, itd.

Navigacijski sustavi su jedan od ključnih dijelova korisničkog sučelja jer omogućavaju korisniku da kroz određeni skup akcija uspijeva ostvariti svoj cilj. Dobri navigacijski sustavi korisniku jasno prikazuju strukturu hijerarhije informacija te mu prikazuju njegovu trenutnu lokaciju. Globalna, lokalna i kontekstualna su najčešće korištene vrste navigacija. Globalna navigacija je prisutna na svakoj stranici kroz cijelo korisničko sučelje. Lokalna navigacija omogućuje korisnicima da istražuju što se sve nalazi na pojedinoj stranici a koristi se kako bi korisnici dobili veći kontekst. Kontekstualna navigacija upućuje korisnike na stranice koje su sadržajno povezane s trenutnom.

Pretražujući sistemi se koriste kako bi se korisnicima pomoglo pronaći određenu informaciju unutar web stranice. Učinkoviti su kada web stranica sadrži mnogo informacija te postoji mogućnost da se korisnik "izgubi". Pretražujući sustavi su u praksi često realizirani tražilicama i filtrima.

### **3.3.3. Izrada prototipa**

Izrada prototipa je važan korak iz različitih razloga. Prvo omogućuju upravljanje dragocjenim resursima poput vremena i novca. Drugo, budući da prototipove možemo brzo izraditi omogućeno je ponavljanje dizajna i približavanje željenom cilju zadovoljenja potreba korisnika.

Mogu se izraditi vodoravni prototipovi (engl. *horizontal prototype*) koji omogućuju modeliranje većeg broja značajki, ali uključuju manji broj funkcionalnosti. Ili se mogu odabratи vertikalni prototipovi (engl. *vertical prototype*) gdje će se detaljno modelirati nekoliko značajki. Prototipovi niske vjernosti (engl. *low fidelity*) imaju malo sličnosti s konačnim dizajnom i u dizajni i u funkciji, dok su prototipovi visoke vjernosti (engl. *high fidelity*) vrlo slični konačnom dizajnu. Dizajner je spremان za izradу prototipova visoke vjernosti kada prototipovi niske vjernosti više ne podržavaju ciljeve dizajna. Drugim riječima, nakon što prođe kroz seriju prototipova male vjernosti i nauči se sve što se može, tada se izgrađuje model koji je bliži konačnoј verziji dizajna.

### **3.3.4. Evaluacija**

U fazi evaluacije se provodi analiza prethodno izrađenih prototipova. Evaluacija je općenito definirana u dvije kategorije. Formativna evaluacija se provodi rano u procesu dizajniranja s prototipovima niske vjernosti, dok se sumativna evaluacija provodi s prototipovima visoke vjernosti ili blizu konačnog sučelja.

Nakon faze evaluacije najčešće slijedi ponovno iteriranje cijelog kupaнog procesa izgradnje dobrog korisničkog iskustva,

# 4. Primjena na sustav Edgar

## 4.1. Definiranje zahtjeva

Edgar je informacijski sustav koji se koristi za testiranje studenata. Sustav mogu koristiti svi, a ciljna skupina su STEM fakulteti. Glavna značajka Edgara je mogućnost ispitivanja proizvoljnog programskog jezika. Ima dvije vrste korisnika: studente i nastavnike.

U svrhu boljeg razumijevanja korisničkih zahtjeva i potreba, napravljeno je istraživanje u obliku dva upitnika i intervjuja. Prvi upitnik je bio namijenjen korisnicima s ulogom nastavnika a drugi za korisnike s ulogom studenta. Upitnik za nastavnike se sastojao od 25 pitanja:

- 2 demografska pitanja,
- 2 pitanja o učestalosti korištenja,
- 6 pitanja o funkcionalnostima,
- 6 pitanja o dojmu dizajna,
- 2 pitanja opisa praktičnog iskustva provođenja predmeta i izrade ispita,
- 3 pitanja o konkurenciji,
- 4 pitanja ocijenjivanja pojedinih svojstava sustava.

Razlike u upitnicima su bile kod pitanja učestalosti korištenja gdje su studenti trebali odgovoriti na 3 pitanja. Zatim, studenti su samo trebali opisati praktično iskustvo prilikom rješavanja ispita te su trebali ocijeniti 7 svojstava sustava. Također, u svrhu dobivanja ažurnijih podataka, studenti koji u trenutku rješavanja upitnika nisu pohađali predmet kroz sustav Edgar su bili upućeni na rješavanje kratkog demo ispita. Upitnici su provedeni preko platforme Google Forms. Upitnike su rješavali korisnici koji su već bili upoznati sa sustavom no sučelje za nastavnike vrlo je složeno pa su intervjuji provedeni kako bi se dobio uvid u korisničko iskustvo novih korisnika u ulozi nastavnika. Intervjuirane osobe su

kroz intervju morale obaviti tri zadatka. Trebali su napraviti jedan ispit i do dati jednu osobu na predmet a potom se ulogirati u studentski račun i riješiti prethodno napravljeni ispit. Dva intervjeta su napravljena uživo a jedan online.

Upitnik za profesore ispunilo je 10 korisnika. Među sudionicima je bila samo jedna ženska osoba i 6 sudionika je u životnoj dobi između 41-50. Svi koriste sustav zbog mogućnosti ispitivanja studenata. Sukladno time, istaknute funkcionalnosti su stvaranje ispita i stvaranje zadataka. U tablici 4.1 se vidi da osim što funkcionalnosti stvaranja testova i zadataka koriste svi sudionici, to su ujedno i funkcionalnosti gdje je najveća potreba za promjenom. Za funkcionalnost tutoriala tablica 4.1 pokazuje da je 10% sudionika navelo ju učestalo koristi, 10% ju nikada ne koristi, niti jedan sudionik ne misli da je funkcionalnost neintuitivna te nitko ne misli da ju je potrebno mijenjati. Osobni dojmovi dizajna korisničkog sučelja su pozitivni a najčešće korištene riječi za njegov opis su "*dobar*" i "*minimalizam*". Tri sudionika su istaknula kako izgled korisničkog sučelja ima mesta za poboljšanje. Neintuitivnost sučelja je jedino negativno svojstvo dizajna gdje se većina korisnika slaže a i učestalo je spomenuti problem kroz upitnik. Slika 4.1 pokazuje da su ocjene svojstava sučelja visoke. Prosječna ocjena jednostavnosti je 3.2, upotrebljivosti 3.7, jasnoća sadržaja 3.4 a sustav Edgar općenito 3.8.

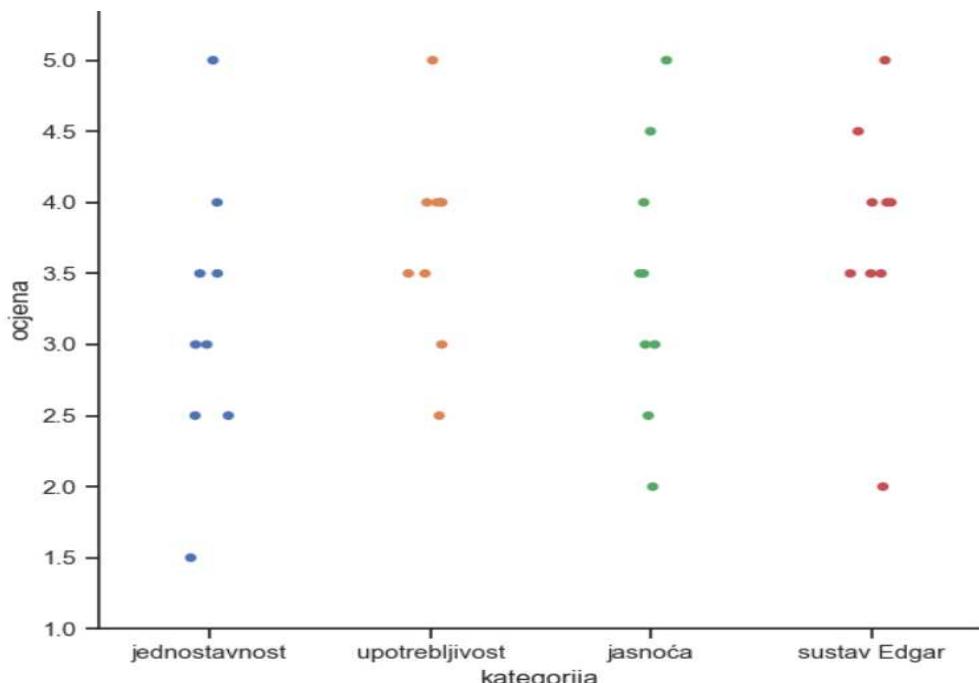
Upitnik za studente ispunio je 21 korisnik. Među sudionicima je 13 ženskih i 8 muških osoba. 20 sudionika su studenti 2. ili 3. godine a samo jedan 1. godine studija; 5 korisnika je u trenutku ispunjavanja upitnika pohađalo predmet preko sustava Edgar a 16 nije. Samo jedan sudionik naveo je da koristi sustav izvan potreba polaganja predmeta. Tablica 4.3 prikazuje rezultate tri nezavisna pitanja što znači da su sudionici mogli odgovoriti da funkcionalnost nikada ne koriste a ujedno nikada nisu ni čuli za nju. Tablica 4.3 pokazuje kako je najučestalija korištena funkcionalnost rješavanje ispita a najmanje korištena code snippets za koju ujedno 13 sudionika niti ne zna da postoji. Za funkcionalnost tutoriala tablica 4.3 pokazuje da je 29% sudionika navelo da ju učestalo koristi, 48% ju nikada ne koristi, a 29% ni ne zna da postoji. Osobni dojmovi dizajna korisničkog sučelja su pozitivni, 10 sudionika je navelo da im se sučelje sviđa, 4 da je jednostavno, 2 da je intuitivno, 3 da ima prostora za poboljšanje a 2 da im se ne sviđa. Tablica 4.4 pokazuje kako većina sudionika smatra da su svojstva navigacije, količine klikanja i intuitivnosti prikladne. Studenti su dali izrazito visoke ocjene sustavu Edgar u svim kategorijama. Prosječna ocjena za korisnost je 4.2, pouzdanost 3.6, upotrebljivost 4.2, jednostavnost korištenja 4.6, jasnoća sadržaja 4.6, vizualni izgled 3.6, a sustav Edgar općenito 3.9.

	Stvaranje testova	Stvaranje zadatka	Unošenje studenata	Tutorial	Code snippets
Učestalo korištena funkcionalnost	10 (100%)	10 (100%)	1 (10%)	1 (10%)	0 (0%)
Nikada korištena funkcionalnost	0 (0%)	0 (0%)	0 (0%)	1 (10%)	2 (20%)
Neintuitivna funkcionalnost	3 (30%)	4 (40%)	1 (10%)	0 (0%)	0 (0%)
Želja za promjenom	2 (20%)	3 (30%)	1 (10%)	0 (0%)	0 (0%)

**Tablica 4.1:** Broj i udio odgovora na pitanje s obzirom na funkcionalnost, N=10

	Da	Ne
Otežana navigacija	4	6
Previše klikanja	2	8
Neintuitivnost	6	4

**Tablica 4.2:** Kategorizacija odgovora o svojstvima dizajna sučelja, N=10



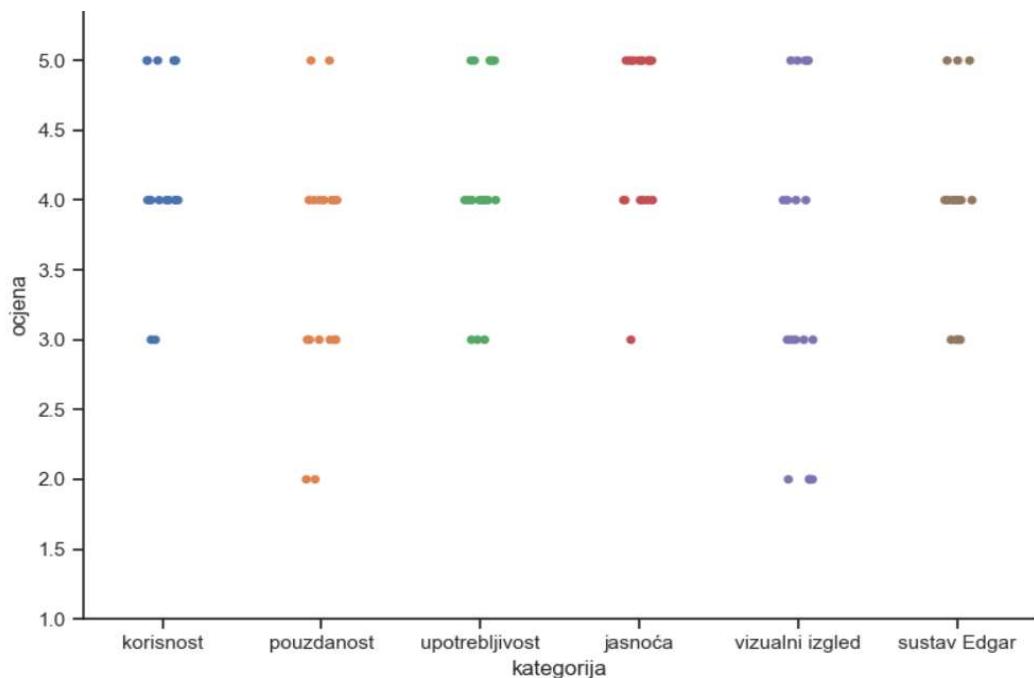
**Slika 4.1:** Distribucije ocjena korisničkog sučelja na skali od 1 do 5, N=10

	Rješavanje ispita	Rješavanje tutoriala	Exercises	Pregled statistika	Code snippets
Učestalo korištene funkcionalnosti	21 (100%)	6 (29%)	10 (48%)	12 (57%)	1 (5%)
Nikada korištene funkcionalnosti	0 (0%)	10 (48%)	6 (29%)	7 (33%)	18 (86%)
Nepoznate funkcionalnosti	0 (0%)	6 (29%)	0 (0%)	3 (14%)	13 (62%)

**Tablica 4.3:** Broj i udio odgovora na pitanje s obzirom na funkcionalnost, N=21

	Da	Ne
Otežana navigacija	1	20
Previše klikanja	3	18
Neintuitivnost	1	20

**Tablica 4.4:** Kategorizacija odgovora o svojstvima dizajna sučelja, N=21



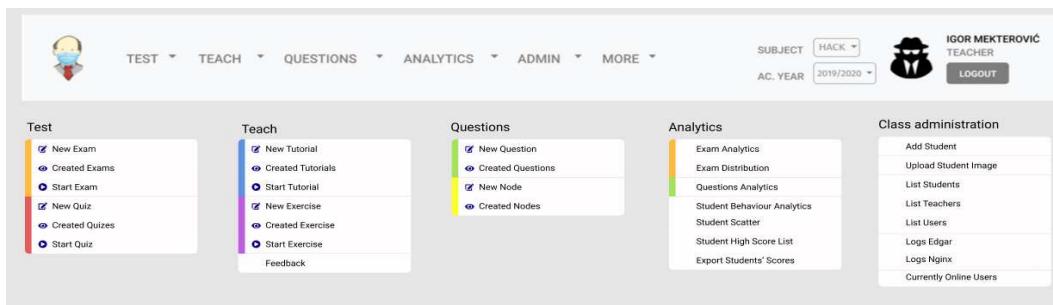
**Slika 4.2:** Distribucije ocjena korisničkog sučelja na skali od 1 do 5, N=21

Intervju je obavljen s tri korisnika. Sva tri sudionika imala su pozitivna inicijalna mišljenja o dizajnu sučelja za nastavnike. Na početku intervjeta sudionici su imali proizvoljnu količinu vremena kako bi istražili i upoznali se sa svim funkcionalnostima sustava. Nakon upoznavanja sa sustavom, prvi zadatak im je bio napraviti jedan ispit s dva pitanja od kojih jedno mora zahtijevati programsko rješenje. Prvom sudioniku je za izradu ispita trebalo otprilike 50 minuta, drugom 70 minuta a trećem također otprilike 70 minuta. Sudionici su tijekom izvršavanja zadatka bili frustrirani. Bez obzira na prethodno upoznavanje sa sustavom nisu mogli pronaći što su htjeli, forme za izradu ispita i zadatka su im bile zbunjujuće te ih je količina podataka obeshrabrilala a slijed radnji koji su trebali napraviti im je bio neintuitivan. Sve duljim prolaskom vremena su gubili samopouzdanje u sebe i pouzdanost u sustav što će napraviti izvršavanjem neke radnje. Imali su česte zamjerke za nekonzistentnost, nelogičnost položaja te veličine upravljačkih elemenata kao što su gumbi. Sudionici su slično reagirali i na drugi zadatak, unos studenta na predmet. Svakome je trebalo otprilike 20 minuta da bi unio jednog studenta. Problem su im stvarale specifikacije i ograničenja vezana za unos svih potrebnih podataka. Zadnji zadatak, rješavanje ispita u sučelju za studente svi su sudionici riješili brzo, svakome je trebalo 5 minuta. Sučelje za studente im je bilo mnogo jednostavnije i intuitivnije za korištenje.

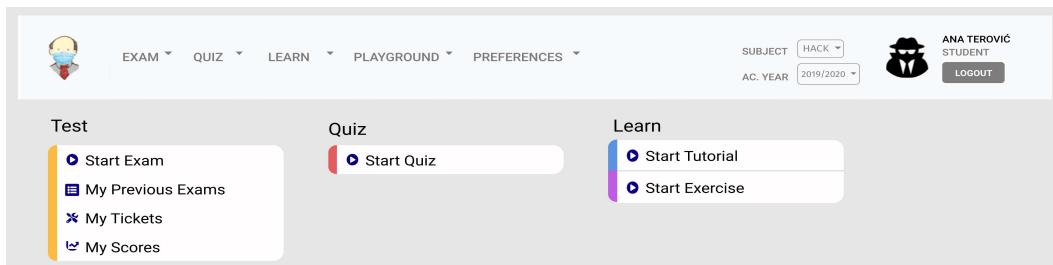
## 4.2. Izrada alternativnih dizajna

Prijedlog redizajna bazirat će se na analizi podataka dobivenih iz provedenih upitnika i intervjuja. Korisnici u ulozi nastavnika najčešće koriste funkcionalnosti vezane za izradu ispita i pitanja a to je ujedno i područje gdje je tražena najveća količina promjena. Kroz intervjuje je uočena problematika korištenja sustava u ulozi nastavnika za nove korisnike. Upitnici koje su ispunjavali studenti nisu pokazivali na područja s izrazitim problemima. Zbog prethodno navedenih informacija, redizajn će se fokusirati na sučelje za nastavnike, specifično forme za izradu ispita i pitanja. Uz njih, napravljen je i prijedlog za navigaciju kako bi se sa strane nastavnika povećala intuitivnost snalaženja kroz sučelje a za studente povećala svjesnost o raznolikosti sadržaja.

Slike 4.3 i 4.4 prikazuju alternativni dizajn navigacije. Labele su promijenjene u izraze koji su učestalo korišteni u drugim sustavima te su iste radnje nazvane identično. Pojedinačne kategorije funkcionalnosti koje nude mogućnost obavljanja raznih radnji kao što su ispiti, kvizovi, itd. kategorizirani su prema bojama kako bi korisnicima bilo lakše uočiti koje sve kategorije postoje. S navigacije za nastavnike su pomaknute funkcionalnosti koje korisnici ne koriste.



Slika 4.3: Alternativni dizajn navigacije u ulozi nastavnika



Slika 4.4: Alternativni dizajn navigacije u ulozi studenta

Forma za izradu ispita i pitanja je prilagođena sukladno načelu progresivnog otkrivanja. Sve funkcionalnosti koje nudi pojedina forma su ostale ali su raspoređene u njima prikladne kategorije. Sam izgled formi nije izrazito promijenjen kako bi već postojeći korisnici mogli nastaviti koristiti sučelje bez velikih prilagodbi.

**Slika 4.5:** Alternativni dizajn forme za ispite, izgled pri otvaranju stranice

**Slika 4.6:** Alternativni dizajn forme za pitanja, izgled pri otvaranju stranice

## EXAM

**Description**

No.	Title	Password	Available From	Available Until
<input type="text"/>	<input type="text"/> Description	<input type="text"/>	<input type="text"/>	<input type="text"/>

---

**Controls**

Global	<input checked="" type="checkbox"/> Off	Show Solutions	<input checked="" type="checkbox"/> Off	Use in Stats	<input checked="" type="checkbox"/> Off
Public	<input checked="" type="checkbox"/> Off	Hint Results	<input checked="" type="checkbox"/> off	Score Ignored	<input checked="" type="checkbox"/> Off

---

Specifications		Time		Files	
No. of Q's	Max Runs	Duration (s)	Question Pace	Upload File	
<input type="text"/>					
Max Score	Pass %	Review Time (min)	Test Pace	Upload Limit	<input type="text"/>
<input type="text"/>					

---

**Nodes**

Add Node	<input type="text"/>	Filter Node Name	<input type="text"/>	<input type="button" value="Fetch"/>
10888	Demo	NodeTest	<input type="button" value="Add"/>	
10770	Demo	Vulnerable node	<input type="button" value="Add"/>	

---

**Exam Parts**

#	Ordinal	Node type and name	Min questions	Max questions	Pass percentage	Grading model	
1	0	Demo(5 qs): NodeTest	<input type="text"/> 0	<input type="text"/> 1	<input type="text"/> 0.0	<input type="text"/> 0/0/0	<input type="button" value="Delete"/>
2	0	Demo(4 qs): Vulnerable node	<input type="text"/> 0	<input type="text"/> 1	<input type="text"/> 0.0	<input type="text"/> 0/0/0	<input type="button" value="Delete"/>

Slika 4.7: Alternativni dizajn forme za ispite u potpunosti otvoren

**QUESTION**

**Question**

Please, answer my clever question...  
Whitespace placeholders (please C/P): ->

**Tags**  
Select Some Options

Add attachment Choose file Browse

**Preview:**  
Please, answer my clever question...  
Server-side render(formulas)

**Solution**

Language: C (GCC 9.2.0) Add attachment Choose file Browse Off Mark Whitespace

Prefix: 1

Main:  
1  
2  
3  
4  
5  
6

Suffix: 1

**Test**

Test item 1 Constraints for test item 1

Toggle #1 Deduct(%) when incorrect: 100.00

1. Type:  1. Fixed  2. Random

2. Settings

Input: 1

Output: 1

3. Grading options

Allow different elements order (NOT TESTED)  
 Case insensitive  
 Trim trailing whitespaces

Regular expression (overrides/ignores the former three options):

Comment: 1

Add new test item

**Comment**

Question comment (optional; GH markdown)

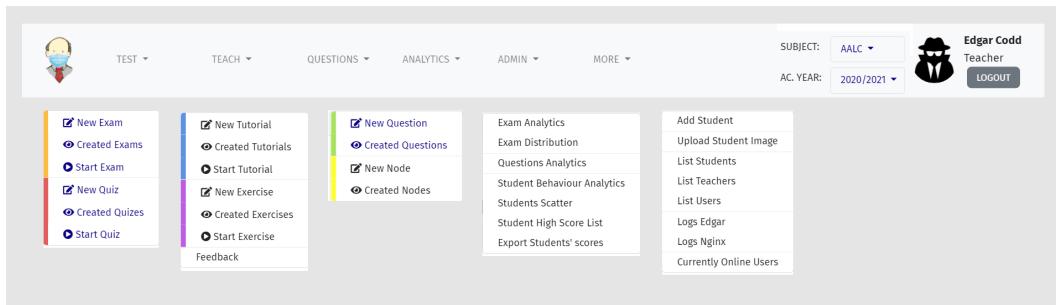
Comment preview:

Run All Tests Clone Clone Up Validate Generate Save

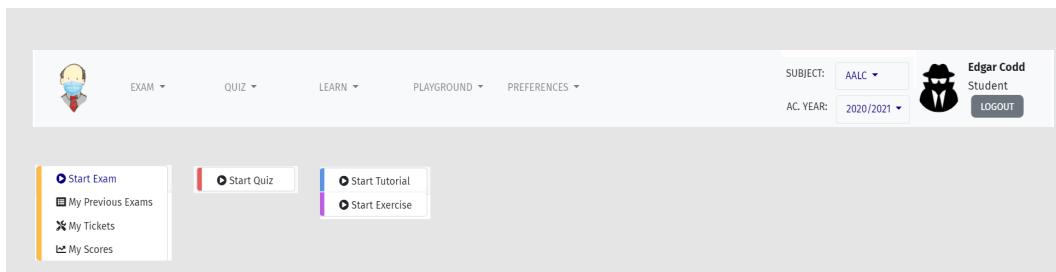
**Slika 4.8:** Alternativni dizajn forme za pitanja u potpunosti otvoren

### 4.3. Izrada prototipa

U sklopu završnog rada implementiran je prototip za navigaciju.



Slika 4.9: Implementirani prototip u ulozi nastavnika



Slika 4.10: Implementirani prototip navigacije u ulozi studenta

## **4.4. Evaluacija**

Evaluacija će biti napravljena s obzirom na tri predložena alternativna dizajna. Promjenom labela navigacije se povećala vanjska dosljednost sustava jer su upotrebljeni izrazi koji se učestalo koriste u drugim sustavima. Povećanjem vanjske dosljednosti ujedno se povećala i pristupačnost sustava za nove korisnike koji će se moći jednostavnije navigirati kroz sustav prepoznajući značenje labela. Unutarnja dosljednost se povećala jer su korištene iste labele kako bi se označile iste akcije. Ova promjena može rezultirati povećanjem samopouzdanja korisnika prilikom korištenja sustava jer će samostalno moći predvidjeti sljedeći korak. S navigacije za nastavnike su pomaknute mogućnosti koje korisnici ne koriste povećavajući omjer signala i buke te ugađajući pravilu 80/20. Forme za izradu ispita i zadataka primjenjuju načelo progresivnog otkrivanja. Forme dijele sličan izgled te su gumbi za akciju postavljeni u dolnjem desnom kutu, mjestu predloženom Guttenbergovim dijagramom.

Mana napravljene analize se nalazi u provedenom istraživanju, točnije upitnicima. Uzorak korisnika koji je sudjelovao je poprilično mali, pogotovo sa strane nastavnika. Iako je bilo moguće raspoznati neke probleme oni su ostali izrazito generalni. Svaki od 10 sudionika imao je drugačiju primjedbu na istu funkcionalnost ne omogućavajući bolju filtraciju specifičnijeg problema čijim rješavanjem bi rezultiralo najvećim poboljšanjem korisničkog iskustva. Uzorak studenata koji su sudjelovali su uglavnom bili studenti 2. ili 3. godine fakulteta a većina njih nije koristila sustav u tekućem semestru što je rezultiralo primjedbama na već riješene probleme. Osim uzorka ispitanika, upitnici su bili napravljeni bez prijašnjeg iskustva izrade takvih upitnika zbog čega su se dogodile početničke pogreške.

## **4.5. Prijedlog za budući razvoj**

Budući razvoj treba biti fokusiran na poboljšanju intuitivnosti i dosljednosti sustava. Korisničko sučelje u ulozi nastavnika je izrazito složeno te postoji granica do koje se neintuitivnost za nove korisnike može riješiti no uvođenjem kvalitetnih i detaljnih uputa, što je česta praksa u ostalim kompleksnim sustavima, taj problem bi se mogao poprilično poboljšati. Pažljivim i detaljnim prakticiranjem dosljednosti kroz cijeli sustav izrazito bi se poboljšala sveukupna kvaliteta sustava a pogotovo intuitivnost i povjerenje korisnika u sustav.

## 5. Zaključak

Jedina dodirna točka korisnika i sustava je korisničko sučelje. Korisnička sučelja se razvijaju kako bi korisnici mogli što lakše i elegantnije koristiti funkcionalnosti koje im web-aplikacija nudi. Zato je izgradnja dobrog korisničkog sučelja jako važan dio izgradnje dobre web-aplikacije bez obzira na kompleksnost sustava.

Cilj ovog završnog rada je napraviti sažeti pregled predloženih konvencija i praksi koje doprinose vizualnom izgledu i iskustvu uporabe korisničkih sučelja. Nakon toga, cilj je i prikazati primjenu navedene teorijske podloge na informacijskom sustavu Edgar.

Rad je realiziran u tri dijela. Teorijska podloga je obrađena kroz tematike korisničkog sučelja i korisničkog iskustva. Korisnička sučelja su razrađena opisom procesa izgradnje, pregledom tehnologija, temeljnih načela izgradnje i dobrih praksi pri izgradnji korisničkih sučelja. Korisničko iskustvo je opisano definiranjem dobrog korisničkog iskustva te metodologijom izgradnje dobrog korisničkog iskustva.

Edgar je informacijski sustav koji se koristi za testiranje studenata. S ciljem poboljšanja korisničkog sučelja i korisničkog iskustva u sustavu Edgar provedena je praktična primjena prethodno navedenih znanja. Prvo je napravljeno istraživanje o korisničkom iskustvu studenata i nastavnika u obliku upitnika i intervjuja. Istraživanje je pokazalo da postoji potreba za promjenom korisničkog sučelja za nastavnike, specifično procesa izrade ispita i pitanja. Istraživanjem nije uočena potreba za promjenama korisničkog sučelja za studente. S obzirom na rezultate istraživanja, izrađeni su alternativni dizajni koji su fokusirani na korisničko sučelje nastavnika. Napravljen je redizajn navigacije, forme za ispite i forme za pitanja. Alternativni dizajn navigacije je implementiran u sustav Edgar.

Proces razvoja dobrog korisničkog iskustva je proces koji se treba uzastopno ponavljati. Sustav Edgar će se tako poboljšati ponovnom provedbom poboljšanja korisničkog iskustva a s time i korisničkog sučelja.

# LITERATURA

- [1] Adobe. Adobexd features, 2021. URL <https://www.adobe.com/products/xd/features.html>.
- [2] Angular. Introduction to angular concepts, 2021. URL <https://angular.io/guide/architecture>.
- [3] Harshani Chathurika. Analogous colors and color wheel, 2019. URL <https://uxplanet.org/analogous-colors-and-color-wheel-609a05b5b90e>.
- [4] C Corner. What and why react.js, 2021. URL <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>.
- [5] deque. axe<sup>TM</sup> - the standard in accessibility testing, 2021. URL <https://www.deque.com/axe/>.
- [6] DEV. Best css frameworks in 2021, 2021. URL [https://dev.to/theme\\_selection/best-css-frameworks-in-2020-1jjh](https://dev.to/theme_selection/best-css-frameworks-in-2020-1jjh).
- [7] MDN Web Docs. Javascript, 2021. URL <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [8] Therese Fessenden. A theory of user delight: Why usability is the foundation for delightful experiences, 2017. URL <https://www.nngroup.com/articles/theory-user-delight/>.
- [9] Figma. Design features, 2021. URL <https://www.figma.com/design/>.
- [10] Norbi Gaal. Ux psychology go hand in hand— how gestalt theory appears in ux design, 2017. URL <https://uxdesign.cc/ux-psychology-go-hand-in-hand-how-gestalt-theory-appears-in-ux-design-18b727>
- [11] Web Hypertext Application Technology Working Group. Html living standard, 2021. URL <https://html.spec.whatwg.org/multipage/>.

- [12] Shawn Hickey i Michael Satran. *Overview of the User Interface Development Process*. Microsoft Docs, 2018.
- [13] jQuery. jquery api, 2021. URL <https://api.jquery.com>.
- [14] William Lidwell, Kritina Holden, i Jill Butler. *Univerzalna načela dizajna*. Rockport Publishers, Inc., 2010.
- [15] Srinivas V. Mondava. Process in user interface development. Magistarski rad, University of Montana, 1998.
- [16] Peter Morville. User experience design, 2014. URL [http://semanticstudios.com/user\\_experience\\_design/](http://semanticstudios.com/user_experience_design/).
- [17] Peter Morville i Louis Rosenfeld. *Information Architecture for the World Wide Web*. O'Reilly Media, Inc., 2002.
- [18] Neoteric. Single-page application vs. multiple-page application, 2016. URL <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>.
- [19] Jakob Nielsen. Jakob's law of internet user experience, 2017. URL <https://www.nngroup.com/videos/jakobs-law-internet-ux/>.
- [20] Jakob Nielsen. 10 usability heuristics for user interface design, 2020. URL <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [21] Anton Nikolov. Design principle: Consistency, 2017. URL <https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f>.
- [22] NNgroup. Don norman: The term "ux", 2016. URL <https://www.youtube.com/watch?v=9BdtGjoIN4E>.
- [23] Don Norman i Jakob Nielsen. The definiton of user experience (ux), 2021. URL <https://www.nngroup.com/articles/definition-user-experience/>.
- [24] Europski parlament. *Pristupačnost internetske stranice Europarl*. URL <https://www.europarl.europa.eu/portal/hr/accessibility>.
- [25] Ben Shneiderman. The eight golden rules of interface design, 2016. URL <https://www.cs.umd.edu/users/ben/goldenrules.html>.

- [26] Andrea May So. Top down lighting bias, 2016. URL <https://www.slideshare.net/AndreaSo2/top-down-lighting-bias>.
- [27] W3 Techs. Usage statistics of javascript libraries for websites, 2021. URL [https://w3techs.com/technologies/overview/javascript\\_library](https://w3techs.com/technologies/overview/javascript_library).
- [28] Tubik. Information architecture basics for designers., 2017. URL <https://uxplanet.org/information-architecture-basics-for-designers-b5d43df62e20>.
- [29] Vue.js. What is vue.js?, 2021. URL <https://vuejs.org/v2/guide/index.html#What-is-Vue-js>.
- [30] W3C. Html css - w3c, 2013. URL <http://www.w3.org/standards/webdesign/htmlcss>.
- [31] Wan Chen Wang i Luiz Moutinho. *Human-Computer Interface in Marketing*. John Wiley Sons, Ltd, 1 2015. doi: 10.1002/9781118785317.weom090132. URL <http://doi.wiley.com/10.1002/9781118785317.weom090132>.
- [32] WAVE. Wave web accessibility evaluation tool, 2021. URL <https://wave.webaim.org>.

**Analiza korisničkog iskustva i dizajn korisničkog sučelja u sustavu  
Edgar**

**Sažetak**

Ovaj rad sadrži pregled predloženih konvencija i praksi koje doprinose vizualnom izgledu i iskustvu uporabe korisničkih sučelja. U radu je napravljena i primjena navedenih konvencija i praksi na informacijskom sustavu Edgar.

**Ključne riječi:** korisničko sučelje, korisničko iskustvo, Edgar

**Title**

**Abstract**

This paper provides an overview of proposed conventions and practices that contribute to a visual appearance and experience of using user interfaces. The paper also includes the application of the mentioned conventions and practices in the Edgar information system.

**Keywords:** user interface, user experience, Edgar