

# IMPLEMENTACIJA PROGRAMSKOG RJEŠENJA ZA UNAPRJEĐENJE POSLOVANJA I UPRAVLJANJA NA PRIMJERU NOĆNOG KLUBA

---

**Mandić, Luka**

**Undergraduate thesis / Završni rad**

**2023**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:225:099685>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-07**



*Repository / Repozitorij:*

[Algebra University - Repository of Algebra University](#)



**VISOKO UČILIŠTE ALGEBRA**

ZAVRŠNI RAD

**Implementacija programskog rješenja za  
unaprjeđenje poslovanja i upravljanja na  
primjeru noćnog kluba**

Luka Mandić

Zagreb, veljača 2023.



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

*U Zagrebu, 3. travnja 2023.*

# Predgovor

Ovaj je rad jedno veliko *hvala* mojoj obitelji na strpljenju.

Temeljem članka 8. Pravilnika o završnom radu i završnom ispitu na preddiplomskom studiju Visokog učilišta Algebra sačinjena je ova

## Potvrda o dodjeli završnog rada

kojom se potvrđuje da student Luka Mandić, JMBAG 0036502920, OIB 76672760185 u šk. godini 2021./2022., studij: Primjenjeno računarstvo - Preddiplomski studij, smjer: Programsko inženjerstvo, od strane povjerenstva za provedbu završnog ispita, dana 01.03.2022. godine, ima odobrenu izradu završnog rada

s temom: **Implementacija programskog rješenja za unaprjeđenje poslovanja i upravljanja na primjeru noćnog kluba**

i sažetkom rada: Cilj ovog rada bio je predstaviti i implementirati programsko rješenje za poslovne i administrativne zahtjeve noćnog kluba na primjeru njegovog specifičnog poslovanja i rada. Metodom anketiranja je provedena analiza prednosti i nedostataka postojećih aplikacija koje su u primjeni. Razvijena je aplikacija koja svojom funkcionalnosti i jednostavnim sučeljem odgovara svim traženim zahtjevima poboljšavajući učinkovitost zaposlenika.

Aplikacija je razvijena koristeći tehnologije ASP.NET i React.js.

Ključne riječi: noćni klubovi, aplikacija, upravljanje, poslovanje, ASP.NET, React.

Mentor je: Bojan Fulanović.

Odobrenjem završnog rada studentu je omogućen upis kolegija "Izrada završnog projekta/Praksa" te je sukladno članku 8. Pravilnika o završnom radu i završnom ispitu dužan najkasnije do početka nastave ljetnog semestra u sljedećoj školskoj godini, uspješno obraniti završni rad uspješnim polaganjem završnog ispita.

U protivnom student može zatražiti novog mentora/icu i temu te ponovo upisati kolegij "Izrada završnog projekta/Praksa" budući da rad koji nije predan i obranjen na završnom ispitu u roku određenom Pravilnikom završnom radu i završnom ispitu prestaje vrijediti. Izrada novog završnog rada se izvodi sukladno rokovima određenima za školsku godinu u kojoj je studentu određen novi mentor/ica i dodijeljen novi završni rad.

Potpis studenta:

Potpis mentora:

Potpis predsjednika  
povjerenstva:

Ova potvrda izdaje se u 4 (četiri) primjerka od kojih 3 (tri) idu kao prilog završnom radu.

## Sažetak

Cilj ovog rada bio je predstaviti i implementirati programsko rješenje za poslovne i administrativne zahtjeve noćnog kluba na primjeru njegovog specifičnog poslovanja i rada. Analiza problema i rješenje te implementacija napravljeni su na stvarnom primjeru noćnog kluba, s mogućim proširenjem i generalizacijom na veći opseg klubova sa sličnim tipom poslovanja.

Pod zahtjevima noćnog kluba uzele su se u obzir administrativne i funkcionalne potrebe, poput upravljanja stanjem skladišta kluba (inventura), upravljanja zaposlenicima i dužnicima, evidencijom prometa kluba, organizacije radnih sati i rasporeda zaposlenika, kao i plaća istih.

U okviru rada, provedena je analiza postojećih metoda koje su u primjeni, točnije, aplikacije Google Sheets i Google Forms te njihovih prednosti i nedostataka. Nastavno na to, razvijena je aplikacija koja bi svojom funkcionalnosti i jednostavnim sučeljem odgovarala svim iznad navedenim zahtjevima, tako poboljšavajući efektivnost obavljanja tih poslova.

Aplikacija je razvijena koristeći tehnologije ASP.NET, SQL Server i React.js.

**Ključne riječi:** noćni klubovi, aplikacija, upravljanje, poslovanje, inventura, ASP.NET, React.

The purpose of this paper was to present and implement an application for nightclub's business needs. A real world example was analysed and a solution was created that could be expanded and generalised to accommodate similar nightclubs.

While gathering requirements for a new implementation, both administrative and functional need were taken into account of which some include inventory tracking, employee management, managing debtors, income tracking, creating schedules and salary calculations.

This paper also analysed the usage of apps Google Sheets and Google Forms for the club's needs and their advantages and disadvantages. Following that, a new application was

implemented to unify all before mentioned needs. The simple to use interface helps the job get done more effectively.

The application was developed using SQL Server database and ASP.NET and React.js frameworks.

**Keywords:** nightclub, application, managing, business, inventory, ASP.NET, React



# Sadržaj

1.	Uvod .....	1
2.	Analiza poslovanja noćnog kluba.....	2
2.1.	Cilj analize.....	2
2.2.	Metodologija.....	2
2.3.	Rezultat analize.....	3
3.	Implementacija .....	6
3.1.	Korištene tehnologije.....	7
3.1.1.	Poslužiteljski moduli .....	7
3.1.2.	Klijentski moduli .....	7
3.2.	Poslužiteljska aplikacija .....	8
3.2.1.	Komunikacija s bazom podataka.....	8
3.2.2.	Autentifikacija i autorizacija .....	9
3.2.3.	Slanje e-pošte.....	10
3.3.	Klijentska aplikacija .....	11
3.3.1.	Responzivnost dizajna .....	11
3.3.2.	Kolačići.....	12
3.3.3.	Čuvanje stanja aplikacije pomoću biblioteke <i>Redux</i> .....	12
3.4.	Funkcionalnosti aplikacije.....	13
3.4.1.	Upravljanje stanjem skladišta.....	13
3.4.2.	Upravljanje zaposlenicima .....	14
3.4.3.	Upravljanje dužnicima.....	15
3.4.4.	Generiranje rasporeda rada.....	16
3.4.5.	Radni sati i plaće .....	18

3.4.6.	Evidencija prometa .....	19
3.4.7.	Grafički prikaz podataka .....	19
4.	Analiza rezultata implementacije .....	20
4.1.	Cilj i metodologija .....	20
4.2.	Rezultat analize.....	20
	Zaključak .....	23
	Popis kratica .....	24
	Popis slika.....	25
	Literatura .....	26
	Prilog .....	27

# 1. Uvod

Noćni klub smatramo dinamičnim poslovnim sustavom, koji je u stalnom zahtjevu za efikasnim i preciznim rješenjima koja bi podržavala njegovo funkcionalno i administrativno poslovanje. Uz generalne zahtjeve, svaki klub ima i svoje specifične metode poslovanja, zbog kojih će vjerojatno, već postojeća rješenja, biti neadekvatna i tek djelomično kompatibilna s zahtjevima tog sustava. Primjerice, određeni noćni klubovi imaju specifično definirano radno vrijeme s raznim restrikcijama, ovisno o više parametara, čije se vođenje i nadzor lako oteža s povećanjem opsega posla i radnika.

Iz tog razloga, bilo je potrebno razviti posebno rješenje, točnije aplikaciju za poslovanje i upravljanje noćnog kluba, koja bi, uz generalne zahtjeve, imala podršku i za određene specifične zahtjeve, zaključene na temelju stvarnog primjera jednog noćnog kluba.

Da bi razvijena i implementirana aplikacija pokrivala zahtjeve u današnjici i stvarnom vremenu, prije svega je potrebna analiza postojećih rješenja, njihovih prednosti i nedostataka. Na temelju rezultata analize, moći će se posložiti prioriteti i način funkcioniranja aplikacije, kao i korisničko sučelje i dizajn koji bi doprinijeli što većoj efikasnosti iste. Takva aplikacija bi trebala biti intuitivna s uključenom pouzdanosti, responzivnosti i lakoćom korištenja.

## **2. Analiza poslovanja noćnog kluba**

### **2.1. Cilj analize**

Cilj analize poslovanja noćnog kluba iz primjera je ostvariti uvid u korištenje programskih rješenja u poslovanju. Prije ikakvog planiranja je nužno saznati koje probleme moramo riješiti i na koje stvari trebamo posebno obratiti pozornost. Također moramo uzeti u obzir činjenicu da iako svi noćni klubovi u suštini imaju slične, skoro pa i identične potrebe, svi se barem po nečem razlikuju.

Upravo te razlike poput opsega ponude noćnog kluba, površine, broja zaposlenih, strukture organizacije i sl. utječu na izradu našeg rješenja. Osim na složenost i arhitekturu aplikacije, utječu i na dizajn sučelja: koja je informacija bitna, na što skrenuti korisničku pozornost, od kojih se loših korisničkih navika zaštititi i sl.

Potrebno je doznati koja se rješenja koriste i u kojem opsegu. Koje su prednosti, a koje su mane trenutnih rješenja. Koji se točno zadaci rješavaju aplikacijama, koliko vremena korisnik u prosjeku potroši na rad s aplikacijama u smjeni te navike korisnika u radu s aplikacijama.

Uz mišljenja jedne skupine korisnika, zaposlenika, zanimaju nas i mišljenja druge skupine - voditelja. Zanima nas što misle o trenutnim rješenjima i koji su njihovi prijedlozi za poboljšanje. Većina funkcionalnosti ovakvog rješenja treba biti skrivena od običnih zaposlenika zbog sigurnosti i održavanja ažurnosti podataka.

### **2.2. Metodologija**

Noćni klub iz primjera broji oko petnaest povremenih zaposlenika. Prikladan način prikupljanja njihovih mišljenja je metoda anketiranja. Anketa je izrađena aplikacijom Google Forms i sakuplja anonimne odgovore zaposlenika o trenutnim aplikacijama koje koriste u poslu.

Anketa je podijeljena u pet skupina pitanja. Svaka skupina pitanja osim posljednje je obavezna. Prva skupina se odnosi na opće informacije o trenutnim aplikacijama, točnije informacije poput koje se aplikacije koriste, koji se zadaci rješavaju tim aplikacijama i koje

su prednosti korištenja aplikacija za te zadatke. Od zaposlenika se u ovoj skupini traži i da procjene vrijeme utrošeno u rad s aplikacijama za vrijeme jedne prosječne smjene.

Druga skupina pitanja daje saznanja koje karakteristike jedne aplikacije zaposlenici smatraju bitnijim od drugih. Karakteristike navedene su brzina učitavanja, jednostavnost unosa, korisnička prilagodba, brzina odgovora aplikacije i dizajn. Zaposleniku je za odgovor ponuđen raspon brojeva od 1 do 10 gdje 1 odražava *nebitno*, a 10 *vrlo bitno*.

Treća skupina pitanja je slična prethodnoj. Zaposlenika se traži da ocjeni prethodno navedene karakteristike trenutnih rješenja koje se koriste brojkom od 1 do 10. Odgovori na ova pitanja i pitanja iz druge skupine će nam pokazati na koje značajke posebno trebamo obratiti pažnju pri izradi novog rješenja.

Četvrta skupina pitanja se odnosi na osobni dojam pojedinačnog korisnika trenutnih aplikacija, tako ostavljajući uvid u neke specifičnosti.

Peta i posljednja skupina pitanja je neobavezna skupina pitanja gdje tražimo zaposlenika da izjasni svoj prijedlog ako ga ima. Također, da napomene ako smatra da smo u izradi ankete propustili neko bitno pitanje.

Uz anketu je odrađen i uvid u postojeće aplikacije i navike korisnika.

## **2.3. Rezultat analize**

Rezultati analize u prvoj skupini pitanja pokazali su da su primarni i najčešći zahtjevi u poslovanju vođenje obračuna, stanja skladišta i prometa, na kraju rada zaposlenika (korisnika) na smjeni. Također, vrlo je bitno imati odvojene uloge korisnika: zaposlenik i voditelj. Uz to, česti zahtjevi specifični za ovaj primjer noćnog kluba su vođenje „knjige duga“ i biranje smjene.

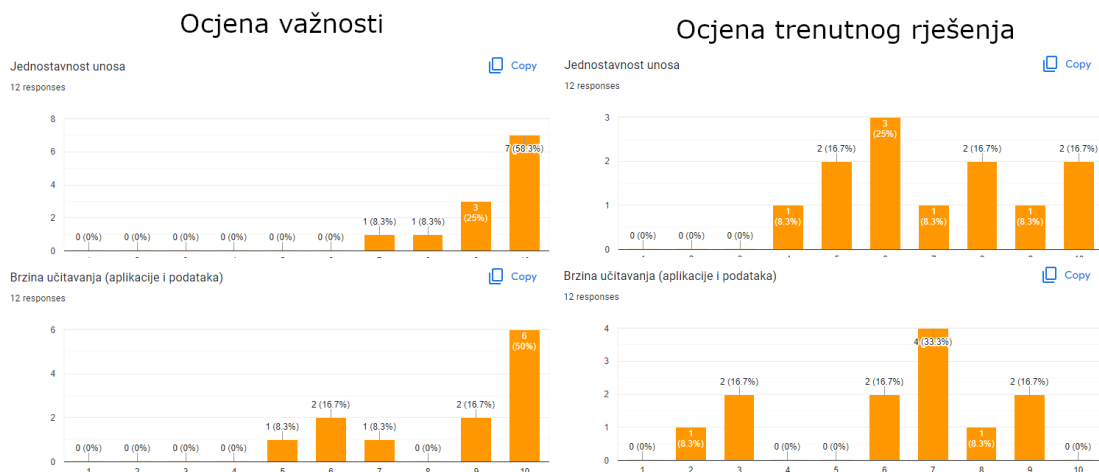
Kao prednosti, navedene su brzina i jednostavnost korištenja, te raspodijeljenost i dostupnost sustava, budući da trenutne aplikacije omogućavaju lako dijeljenje podataka među korisnicima. Informacije bitne za rad sadržane u ovim aplikacijama su stanje skladišta, rasporedi i lako dostupne forme za ispunjavanje obračuna. Korisnici u prosjeku troše 20-30 minuta na korištenje tih aplikacija.

U drugoj skupini pitanja, gdje su istražene najpotrebnije karakteristike aplikacija, ispitane su parametri: brzina učitavanja, jednostavnost unosa, korisnička prilagodba izgleda i načina unosa, responzivnost i dizajn te prilagodba malim ekranima. Od navedenih karakteristika,

jednostavnost unosa pokazala se kao najbitnija gdje je preko 50% ispitanika označilo najvišim prioritetom. Potom slijede karakteristike brzine učitavanja aplikacije i responzivnosti aplikacije (50%). Parametri prilagodbe izgleda aplikacije i dizajna su se pokazale osrednje bitnima.

U trećoj skupini pitanja, gdje su gore navedene karakteristike ispitane na postojećim rješenjima, brzina učitavanja i jednostavnost unosa, pokazano dvije prioritetne karakteristike, ocjenjene su prosječnom ocjenom (6-7), što pokazuje da ima prostora za poboljšanje. Responzivnost postojećih aplikacija je ocjenjena najvišom ocjenom, dok je korisnička prilagodba izgleda i načina unosa, kao i sam dizajn aplikacija u prosjeku najlošije ocijenjen.

Grafički prikaz odgovora zaposlenika vezanih za najbitnije značajke i ocjene tih značajki u trenutnom rješenju je prikazan na slici Slika 2.1. Lijevi stupac prikazuje raspodjelu odgovora za važnost karakteristike, a desni stupac ocjenu tih karakteristika trenutnog rješenja.



Slika 2.1 Grafički prikaz odgovora ankete za dvije ispitanicima najbitnije karakteristike

Ispitivajući osobni dojam korisnika u četvrtoj skupini pitanja, najvećom prednosti smatra se jednostavnost i intuitivnost korištenja postojećeg rješenja. Međutim, kao mane navedene su loša integriranost podataka, prilagodba drugim sustavima i ekranima i nepreglednost. Posljedično se događaju i pogreške pri korištenju aplikacija, koje utječu na istinitost i korisnost podataka koji se bilježe. Od značajki predloženih za bolje ili dodatne funkcionalnosti, navedene su poboljšanje dizajna aplikacije kako bi pomogla u preglednosti i korištenju, poboljšanje jednostavnosti unosa, integriranost podataka kao i integriranost aplikacije i lokacija unosa. Zadovoljstvo postojećih rješenja je po rezultati ankete malo iznad prosječnog.

U petoj, završnoj skupini pitanja, nismo dobili nikakve savjete ni zahtjeve. Razlog tome može biti da zaposlenici ne razmišljaju puno o novim funkcionalnostima ili su zbog neobaveznosti pitanja ostavili prazno ne bi li prije završili s ispunjavanjem forme.

Bez obzira na manjak specifičnih zahtjeva zaposlenika, rezultati iz prethodnih skupina pitanja i osobni uvid u aplikacije i njihov rad su složili kompletnu sliku trenutne situacije u klubu iz primjera. Samim time, imamo sve potrebne informacije za izradu i implementaciju novog rješenja.

### 3. Implementacija

Sagledavši rezultate analize i uočene potrebe, složeni su zahtjevi nove aplikacije. Nova aplikacija mora objediniti funkcionalnosti trenutno korištenih Google Sheets i Google Forms aplikacija. Različite funkcije moraju uvijek biti dostupne radi lakšeg prebacivanja između zadataka. Ovaj zahtjev je riješen odabirom SPA (engl. *Single Page Application*) arhitekture za klijentsku aplikaciju.

SPA tip aplikacije predstavljen je jednom stranicom, koja kontinuirano komunicira s korisnikom aplikacije putem dinamičkog „prepisivanja“ sadržaja trenutne stranice sa sadržajem nove stranice, umjesto učitavanja cjelovitih novih stranica sa servera. Prednosti ovakvog tipa su brzina učitavanja, što posljedično utječe i na brzinu poslovanja koja je ključna u dinamičkom okruženju noćnog kluba. Također, ovakve aplikacije zahtijevaju manju propusnost (engl. *Bandwidth*), jer rade na principu samo jednog učitavanja stranice, pa tako dobro funkcioniraju u područjima s ometenom ili sporom internet vezom. Iskustvo korištenja ovog tipa je uz to za korisnika ugodno, jer se izbjegava proces i pregled učitavanja kompletne stranice, budući da se osvježava samo sadržaj iste. U području implementacijskih detalja, ovaj tip aplikacije je odličan za naknadno proširenje, odnosno dodavanja funkcionalnosti.

Jedna od mana SPA aplikacija su loš performans u području SE optimizacije (engl. *Search Engine Optimization*). Kako naše rješenje nije namijenjeno korisnicima van organizacije, tj. korisnicima bez poveznice, ovu manu možemo zanemariti. Dostupnost cijelog kôda aplikacije u klijentskom pregledniku predstavlja veliki problem u području sigurnosti. Potrebno je posebno pripaziti na validaciju korisničkog unosa i svu poslovnu logiku nad podacima obavljati na serveru. SPA aplikacije u prosjeku troše puno više resursa web preglednika od aplikacija s više stranica. Naše rješenje će imati odvojenu poslovnu logiku od prezentacijskog sloja što uvelike smanjuje „teret“ nad radnom memorijom web preglednika. (Net Solutions, 2021.)

Aplikacija je razvijena sa slojevitom arhitekturom, u vidu da se otvori mogućnost dodavanja mobilne inačice aplikacije nekad u budućnosti. Iako izrada mobilne inačice ne ulazi u opseg ovog rada, posebno je obraćena pažnja na prikaz za male ekrane.



## 3.1. Korištene tehnologije

### 3.1.1. Poslužiteljski moduli

Poslužiteljska je aplikacija izrađena u ASP.NET razvojnom okviru tvrtke Microsoft. ASP.NET se nadograđuje na razvojni okvir .NET s alatima i bibliotekama specifičnim za izradu web aplikacija. Otvorenog je kôda i s velikim brojem korisnika što olakšava otklanjanje problema u razvoju i održavanju.

Od korištenih biblioteka treće strane važno je spomenuti sljedeće:

- Entity Framework Core v6.0.5
- BCrypt.Net v4.0.3
- Serilog v2.11.0
- MailKit v3.3.0

Kasnije u radu će se objasniti svrha navedenih biblioteka.

### 3.1.2. Klijentski moduli

Klijentska aplikacija je izrađena u React.js razvojnom okviru. React.js je trenutno jedan od najkorištenijih okvira za razvoj SPA aplikacija i razvoj web aplikacija općenito. (Statista.com, 2022.) Prednosti tako velike raširenosti je velik izbor biblioteka koje mogu uvelike olakšati i ubrzati razvoj. Neke od biblioteka korištenih u novom rješenju:

- Next.js v12.2.5
- NextAuth.js v4.13.0
- Material UI v5.8.1
- mui-datatables v4.3.0
- react-redux v8.0.2

Korištenjem razvojnog okvira Next.js, nudi se mogućnost implementacije node.js poslužitelja za klijentsku aplikaciju. Implementacijom poslužitelja za klijentski dio štedimo na resursima web preglednika pošto se dio procesiranja obavlja na poslužitelju. Poželjno je i iskoristiti značajku SSR-a (engl. *Server-Side Rendering*), tj. generiranje HTML-a na poslužitelju, što rezultira bržim učitavanjem stranica u pregledniku.

Material UI je razvojni okvir za vizualno oblikovanje aplikacije koji se drži smjernica Material Design-a tvrtke Google. Material Design je skup smjernica, komponenata i alata

koji podržavaju najbolje prakse dizajniranja korisničkih sučelja. Korištenjem jedne biblioteke za dizajn sučelja olakšava održavanje konzistentnosti dizajna kroz različite stranice i komponente aplikacije. Uz to, biblioteka nudi jednostavan način implementacije promjene teme aplikacije u svijetlu odnosno tamnu paletu boja.

Ostale će biblioteke biti pojašnjene kasnije u radu.

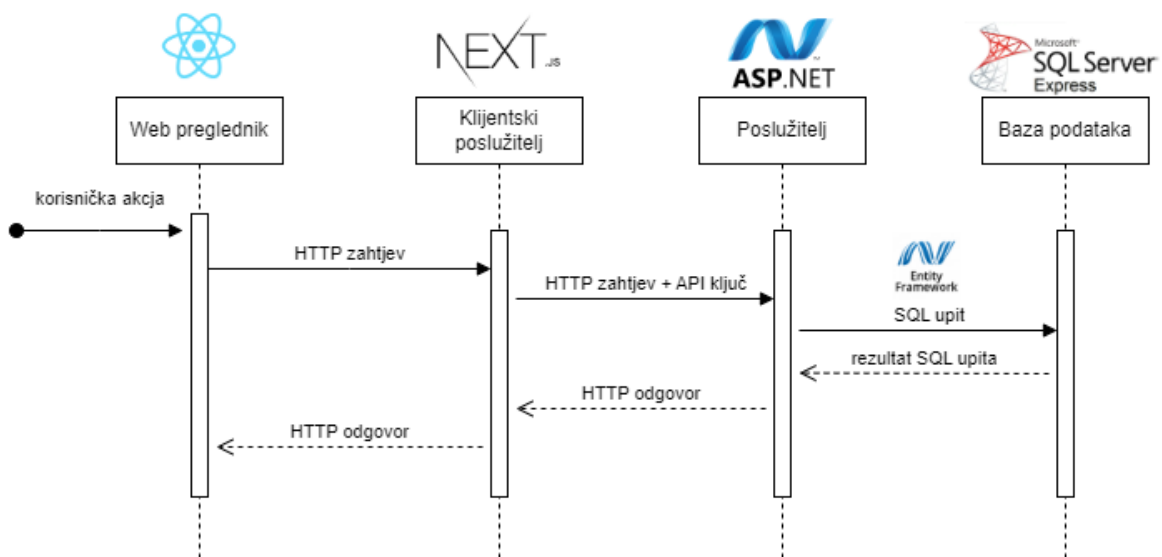
## **3.2. Poslužiteljska aplikacija**

### **3.2.1. Komunikacija s bazom podataka**

Upiti prema bazi podataka se odvijaju isključivo putem poslužiteljskog dijela aplikacije. Radi sigurnosnih razloga, svi upiti prema bazi se moraju kvalitetno ispitati. Samo validacija na klijentskoj aplikaciji nije dovoljna i bilo tko s malo poznavanja programskog jezika JavaScript može zaobići taj tip validacije. Zato je nužno obaviti validaciju i na poslužiteljskoj strani kako bi se očuvala dosljednost podataka.

Upiti prema bazi se šalju pomoću programskog okvira Entity Framework Core (EF Core). EF Core obavlja posao objektno relacijskog mapiranja (engl. *Object-relational mapping*) objekata u .NET programskom okviru i relacija u bazi podataka. Uz to, smanjuje količinu kôda potrebnog za napisati ne bi li se obavljale operacije nad bazom podataka i štiti nas od potencijalnih napada SQL injekcijom (engl. *SQL injection*).

Za svaki uspješan upit prema bazi podataka, poslužiteljska aplikacija klijentskoj vraća HTTP status kôd 200 - OK sa rezultatom upita u JSON formatu. U slučaju da upit prema bazi iz bilo kojeg razloga ne vrati rezultat, poslužitelj vraća HTTP status kôd 400 – Neispravan zahtjev (engl. *Bad request*) s kratkim opisom greške te se cijela greška zapisuje u evidencije (engl. *Log*) aplikacije. Sekvencijalni dijagram jednog zahtjeva od klijentske aplikacije do baze podataka i natrag je prikazan na slici Slika 3.1.



Slika 3.1 Sekvencijalni dijagram zahtjeva klijentske aplikacije prema bazi podataka

### 3.2.2. Autentifikacija i autorizacija

Autentifikacija korisnika je riješena na jednostavan način uspoređivanja zaporke i e-pošte. Radi veće sigurnosti i privatnosti korisnika, sve zaporke prije spremanja u bazu prolaze kroz postupak hashiranja (engl. *Hashing*). *Hashiranje* je postupak koji omogućuje generiranje jedne ili više vrijednosti iz niza teksta pomoću matematičke formule. Formula korištena u poslužiteljskoj aplikaciji se naziva „bcrypt“.

Kod uspješne autentifikacije poslužitelj klijentu vraća HTTP status 200 sa svim podacima o korisniku uključujući i njegove postavke aplikacije. U slučaju neuspješne autentifikacije, klijentskoj aplikaciji se vraća jednu od dvije moguće poruke greške: unesena je kriva zaporka i korisnik ne postoji.

Pri izradi poslužiteljske aplikacije iskorištena je značajka ASP.NET okvira naziva srednji sloj (engl. *middleware*). *Middleware* je dio aplikacije, dio programskog kôda, koji će se izvršiti na svaki nadolazeći zahtjev.

Jedan od funkcionalnih zahtjeva voditelja je kontrola svih akcija nad podacima aplikacije. Ta je funkcionalnost riješena uz pomoć biblioteke Serilog. Biblioteka Serilog nudi jednostavno korištenje i integraciju sa SQLServer bazom podataka. Kreirana je nova tablica u bazi podataka naziva „Log“ u koju Serilog zapisuje informacije. U *Middleware* je dodano zapisivanje svih zahtjeva koji mijenjaju podatke, tj. svi HTTP zahtjevi tipa POST, PUT, PATCH i DELETE. Za svaki se zahtjev bilježi vrijeme, kratki opis zahtjeva, identifikacijski broj korisnika i podaci koji se mijenjaju.

Dodana je i funkcionalnost dohvaćanja svih zapisa iz baze podataka što omogućuje voditeljima praćenje svih akcija i integriteta podataka. Zbog ograničenosti prostora za pohranu na poslužitelju, svi zapisi stariji od mjesec dana se automatski brišu.

Osim evidencije zahtjeva, korisno je postaviti i logiku autorizacije nadolazećih zahtjeva. (Lock, 2021.) *Middleware* u novom rješenju očekuje u svakom zahtjevu parametar „APIKey“, jedinstveni tajni ključ. Ako na poslužitelj dođe zahtjev bez parametra „APIKey“ ili se ključ ne poklapa s očekivanim ključem, daljnja obrada zahtjeva se zaustavlja i poslužitelj vraća HTTP status 403 koji predstavlja zabranjen pristup traženom resursu (engl. *forbidden*). Ovom implementacijom smo osigurali da pristup poslužitelju imaju samo oni s kojima postoji definiran dijeljeni tajni ključ.

Osim što se autorizacija korisnika obavlja na klijentskoj strani, krajnje točke poslužitelja (engl. *endpoint*) koje su namijenjene isključivo voditeljima prolaze kroz dodatnu autorizaciju. Zahtjevi prema tim krajnjim točkama uz već navedeni parametar „APIKey“ sadrže i identifikacijski broj korisnika koji je inicirao zahtjev. Pomoću tog broja se iz baze podataka dohvaća korisnik te provjerava njegova uloga.

### **3.2.3. Slanje e-pošte**

Sve obavijesti u aplikaciji se šalju e-poštom. Svaki korisnik već mora imati adresu e-pošte stoga nije potrebno prijavljivanje na bilokakve nove servise. Noćni klub iz primjera na raspolaganju ima Google Workspace pretplatu. Izrađen je novi korisnički račun u organizaciji koji će slati e-poštu korisnicima. Pošto su svi korisnici u istoj organizaciji, male su šanse da će obavijesti nove aplikacije završiti u neželjenoj pošti.

Integracija slanja e-pošte u novo rješenje je odrađeno putem Gmail API sučelja i biblioteke MailKit. Biblioteci Mailkit je potrebno predati vjerodajnice poput korisničkog imena, tj. adrese e-pošte i lozinke te adresu SMTP poslužitelja. Kao adresa e-pošte je postavljena adresa novokreiranog korisnika za slanje e-pošte korisnika, a SMTP poslužitelj u ovom slučaju je smtp.gmail.com.

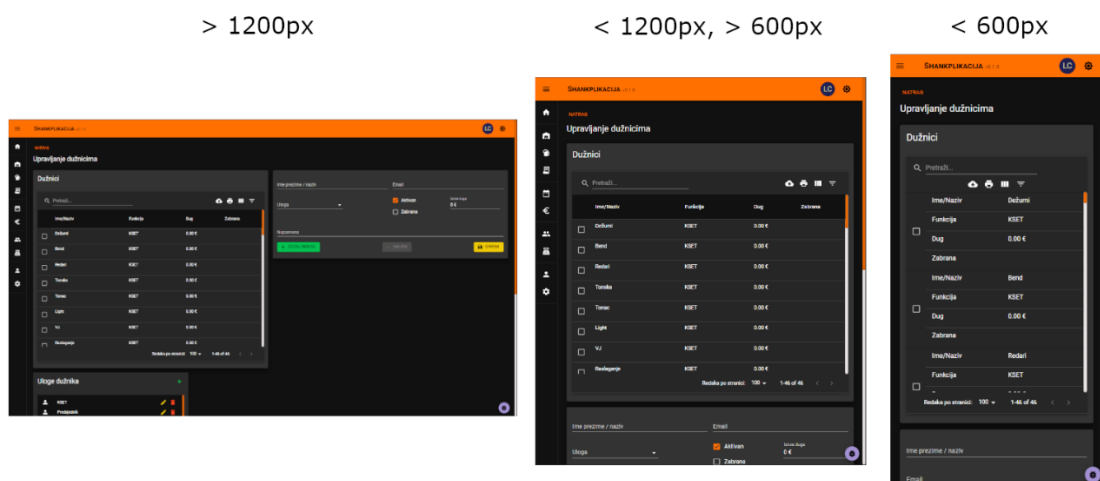
## 3.3. Klijentska aplikacija

### 3.3.1. Responzivnost dizajna

Jedna od čestih primjedbi na trenutna rješenja je slaba prilagodba malim ekranima. Responzivan dizajn predstavlja tehniku u kojoj se dizajn stranice automatski prilagođava širini zaslona korisnika. Pri razvoju je zato nužno uzeti u obzir sve veličine ekrana i pobrinuti se da su elementi stranice konzistentni i prilagodljivi. Problem je riješen koristeći medijske upite (engl. *media queries*) u jeziku CSS.

Jedan od najvećih izazova implementacije responzivnog dizajna je pozicioniranje velikog broja podataka na zaslone malih širina. Jedan od takvih primjera su tablice sa više stupaca. Želimo li sačuvati veličinu teksta i čitljivost, na mali zaslon može stati samo toliko stupaca dok se ostali stupci ne pomaknu van granica zaslona. Ako korisnik želi vidjeti „skriveno“ stupce mora koristiti i vodoravno pomicanje po zaslonu.

Korištenjem biblioteke *mui-datatables* imamo opciju sažimanja tablice u listu kartica s podacima. Tablice se prilagođavaju malim zaslonima transformiranjem svakog retka u karticu s dva stupca: prvi stupac je naziv stupca u tabličnom prikazu a drugi stupac vrijednost ćelije. Primjer kako se elementi stranice prilagođavaju širini zaslona je prikazan na slici Slika 3.2.



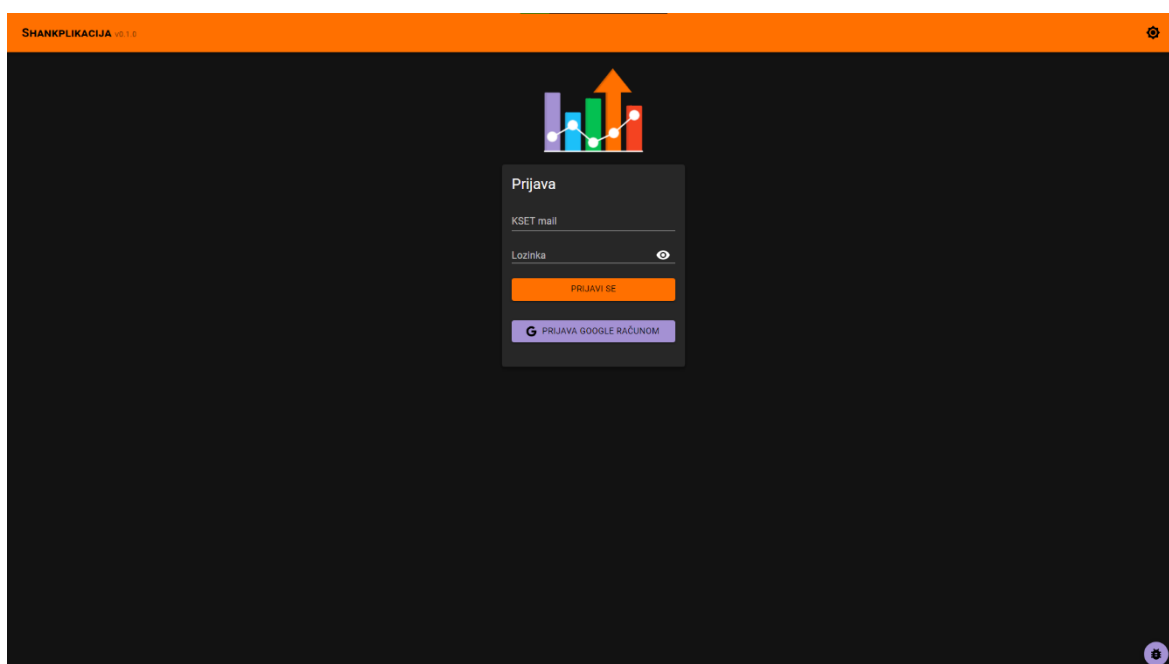
Slika 3.2 Primjer responzivnosti dizajna aplikacije u odnosu na širinu zaslona u pikselima

Sažimanjem tablica smo eliminirali potrebu za vodoravnim pomicanjem i tako ujednačili kretanje po stranici. Mana sažetog prikaza tablice je gubitak lakoće usporedbe podataka u tablici stoga je ova značajka aplikacije dostupna za isključivanje u korisničkim postavkama.

### 3.3.2. Kolačići

Autentifikacija i autorizacija korisnika na klijentskoj aplikaciji se odvija uz pomoć NextAuth biblioteke otvorenog kôda. NextAuth je skup alata za autentifikaciju za aplikacije koje koriste Next.js razvojni okvir. Uvelike olakšava prijavu u sustav i integraciju s drugim davateljima usluga autentifikacije poput Google-a.

Nakon uspješne autentifikacije kreira se kolačić (engl. *cookie*) naziva „next-auth.session-token“. Vrijeme isteka kolačića je trideset minuta. Ako na poslužitelj za klijentsku aplikaciju dođe zahtjev bez „next-auth.session-token“ kolačića, korisnik je preusmjeren na stranicu za prijavu ilustrirana na slici Slika 3.3.



Slika 3.3 Stranica za prijavu korisnika u sustav

U istom tom kolačiću čuvamo i informaciju o ulozu prijavljenog korisnika. Tako možemo lako ograničiti funkcije aplikacije koje nisu namijenjene običnim korisnicima.

### 3.3.3. Čuvanje stanja aplikacije pomoću biblioteke *Redux*

Navigacijom po aplikaciji, nerijetko je potrebno sačuvati stanje aplikacije radi boljeg korisničkog iskustva. Primjerice kada korisnik ne dovrši popunjavanje forme i ode na drugu stranicu, želimo sačuvati unesene vrijednosti kada se korisnik sljedeći put vrati na popunjavanje forme.

Biblioteka Redux nudi alate za čuvanje stanja u aplikaciji. Redux je centraliziran tako da čuva kompletno stanje aplikacije na jednom mjestu. Za očuvanje stanja aplikacije nakon ponovnog učitavanja, moramo se poslužiti lokalnom pohranom (engl. *local storage*) web preglednika.

Redux biblioteka nudi mogućnost implementacije *middlewarea*, točnije, dio kôda koji će se izvršiti na svaku promjenu stanja aplikacije. Naše rješenje na svaku promjenu stanja sprema novu vrijednost stanja aplikacije u lokalnu pohranu web preglednika. Pri učitavanju stranice provjerava se postojanje objekta stanja aplikacije u lokalnoj pohrani i učitava se u aplikaciju. Na taj način korisnik može napustiti stranicu, pa čak i zatvoriti web preglednik, i biti siguran da će stanje aplikacije i svi uneseni podaci biti spašeni.

## 3.4. Funkcionalnosti aplikacije

Sljedeća će poglavlja pobliže opisati funkcionalnosti aplikacije i način korištenja.

### 3.4.1. Upravljanje stanjem skladišta

Za provjeru stanja skladišta koriste se dva tipa inventure – mala i velika. Mala inventura podrazumijeva onu nakon svake smjene, dok se velika obavlja jednom godišnje.

Svaka nova inventura započinje odabirom datuma, tipa inventure i skladišta. Svi ti podaci su zadano postavljeni na datum kada se radi nova inventura, mala inventura i „Glavno skladište“ respektivno. Radi lakšeg snalaženja, prikaz artikala je tabličan s mogućnosti pretraživanja.

Podaci nove inventure se unose u za to predviđenu ćeliju i stupac u tablici. Tablica sadrži tri stupca:

- Artikl
- Puni paketi (količina u paketu)
- Frižider

Zadatak zaposlenika je prebrojati stanje artikala te ih unijeti u točno određene ćelije. Postavljanjem fokusa unosa na polje artikla, redak odabranog artikla ima svjetliju boju od ostalih kao preventivnu mjeru zabunom unesenih podataka.

Nakon što korisnik unese vrijednosti u oba stupca artikla, s desne strane se prikazuje razlika u stanju u odnosu na prošlu inventuru kao što je prikazano na slici Slika 3.4

Artikl	Puni paketi (Količina u paketu)		Frizider
Jana	2	12	18 (-3)
Jamnica		12	

Slika 3.4 Prikaz unosa podataka pri inventuri

Razlika u pravilu treba biti negativan broj, tj. količina prodanih artikala u smjeni. Ako je iz nekog razloga, primjerice krivog unosa u prethodnoj inventuri, razlika veća od nule, korisnik ne može spremiti inventuru bez popunjenog polja Napomena. U tom polju se očekuje od zaposlenika da objasni nepravilnosti u inventuri.

Nakon što su unesene količine za sve artikle zaposlenik može pritiskom na gumb Spremi spremiti novo stanje u bazu podataka. Nakon uspješnog spremanja, na serveru se provjerava postoji li artikl kojeg će ubrzo ponestati. Ako postoje artikli čije je stanje manje od 20% količine artikala u paketu, šalje se obavijest voditeljima putem e-maila. Unosom artikala u bazu, postiže se uniformnost i integritet podataka pa je tako lakše držati i konzistentnost i istinitost podataka. Uz to, poboljšana je preglednost i pretraživanje artikala u skladištu.

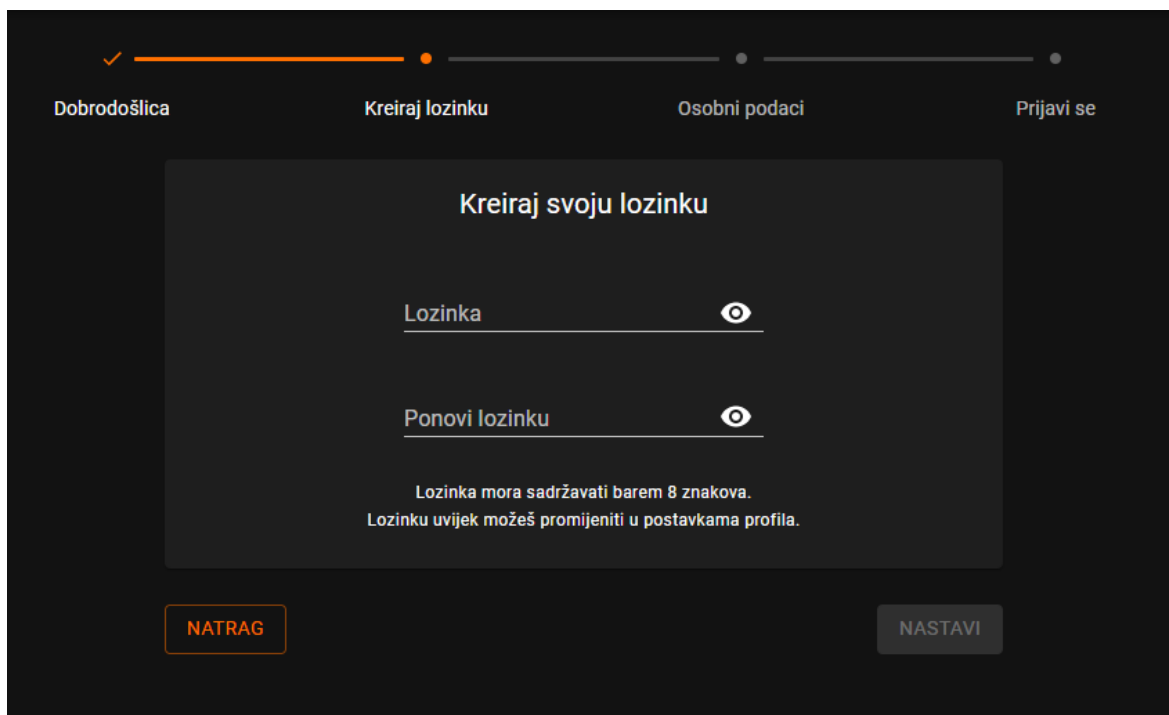
Zbog nerijetkih promjena dobavljača i/ili pakiranja artikla, potrebno je omogućiti jednostavne promjene nad artiklima. Mijenjanje naziva, količine u paketu i svih ostalih atributa artikala je omogućeno isključivo voditeljima kroz upravljačko sučelje.

### 3.4.2. Upravljanje zaposlenicima

Voditelji kroz upravljačko sučelje mogu upravljati svim korisnicima koji imaju pristup aplikaciji. Dodavanje novih korisnika radi na principu pozivnice. Voditelj kroz aplikaciju šalje e-mail novom zaposleniku sa poveznicom na stranicu za registraciju. Na stranici za registraciju novi korisnik unosi potrebne osobne podatke i kreira svoju zaporku. Kreiranje zaporke je prikazano na slici Slika 3.5. Nakon uspješne registracije, korisnik je dodan u bazu podataka i može se prijaviti u aplikaciju.

Obični zaposlenici na stranici svog profila imaju mogućnost promjene zaporke i datuma sanitarne iskaznice dok voditelji mogu brisati korisnike i korisničke podatke, ali i dodavati uloge tj. ovlasti u aplikaciji.



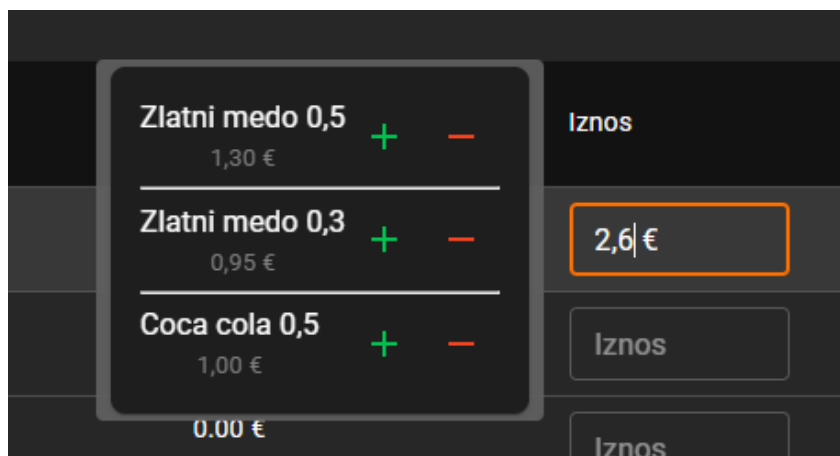


Slika 3.5 Prikaz kreiranja lozinke pri registraciji novog korisnika

### 3.4.3. Upravljanje dužnicima

Jedna od specifičnosti u poslovanju kluba iz primjera je uzimanje artikala na dug. U organizaciji kluba određene osobe koje obavljaju neku funkciju imaju mjesečni budžet koliko pića smiju popiti besplatno. Ako osoba premaši svoj mjesečni budžet, ona i dalje može uzimati pića bez plaćanja ali se dug zbraja i prenosi u sljedeći mjesec.

Zaposlenik na kraju svake smjene popunjava tablicu sa osobama koje imaju svoj budžet. Za unos je potreban iznos u eurima, tj. vrijednost pića kojeg je osoba uzela. Artikli koji se najčešće uzimaju „na dug“ su piva. Kako bi ubrzali proces unošenja iznosa uglavnom različitih količina istih artikala, kraj polja za unos se pojavljuje mali modalni prozor s nekoliko najčešće korištenih artikala i gumbima „+“ za dodavanje i „-“ za oduzimanje prikazan na slici Slika 3.6. Pritiskom na neki od gumba, polju za unos se pridodaje, odnosno oduzima, vrijednost jednog artikla. Ovim je načinom posebice ubrzan unos korisnicima dodirnih ekrana.

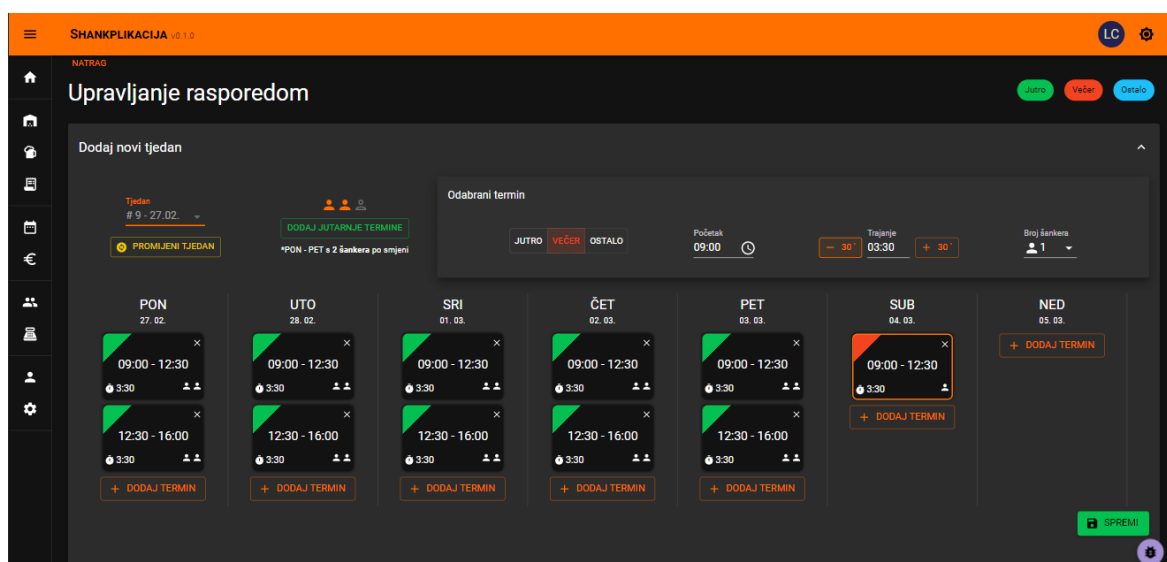


Slika 3.6 Prikaz modalnog prozora za brzo dodavanje tj. oduzimanje iznosa posudbe

Nakon popunjavanja svih potrebnih iznosa i pritiska na gumb „Spremi“, nove se vrijednosti zapisuju u bazu podataka te se osobama koje su imale novi unos šalje obavijest e-mailom. E-mail sadrži informacije kada i koliko je osoba uzela pića i koliko iznosi novi trenutni dug.

### 3.4.4. Generiranje rasporeda rada

Generiranje rasporeda je proces od nekoliko koraka. Prije svega, voditelj mora definirati radne dane i smjene u tjednu. Na raspolaganju mu je sučelje s mogućim odabirom nekog od nadolazećih tjedana, gumb za dodavanje standardnih jutarnjih smjena te kartica za uređivanje pojedine smjene prikazano na slici Slika 3.7. Kartica za uređivanje smjene nudi opcije uređivanja tipa smjene, početka i trajanja smjene i broja potrebnih zaposlenika.



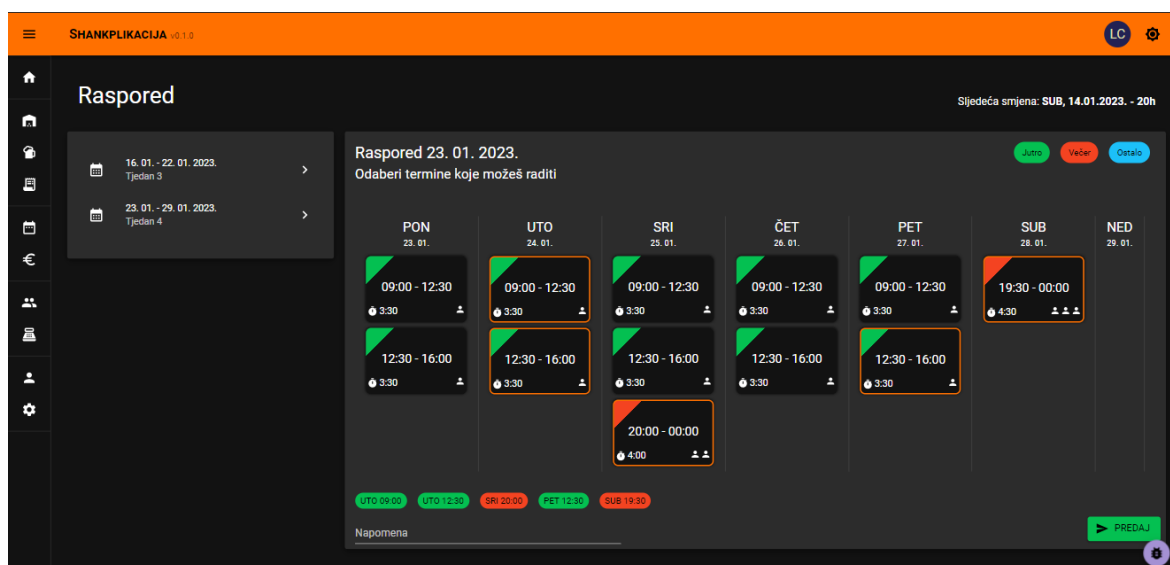
Slika 3.7 Prikaz sučelja za izradu tjednog rasporeda

Radi jednostavnijeg vizualnog prikaza, dodane smjene se dinamički prikazuju u tabličnoj strukturi. Svaki stupac sadrži svoju oznaku dana u tjednu, kronološki poredane smjene u tom danu i gumb „Dodaj smjenu“ koji dodaje novu smjenu u tom danu.

Svaka kartica smjene ima jasno naznačeno trajanje, broj potrebnih zaposlenika i tip smjene naznačen odgovarajućom bojom gornjeg lijevog kuta. Odabirom kartice smjene, iscrtava se narančasti obrub oko odabrane smjene i može se uređivati kroz karticu za uređivanje smjene.

Nakon popunjavanja svih smjena, i neobvezne napomene, klikom na gumb „Spremi“ sprema se novi raspored. Podaci se zapisuju u bazu podataka i šalje se obavijest svim aktivnim zaposlenicima putem elektroničke pošte da je dostupan novi raspored za popuniti.

Nakon što je novi raspored objavljen, zaposlenik ima mogućnost prijaviti se za rad. Navigacijom na stranicu rasporeda, prikazuje se lista nepopunjenih tjednih rasporeda. Odabirom rasporeda, prikazuje se tablični prikaz smjena u tjednu kao i kod kreiranja novog rasporeda prikazan na slici Slika 3.8. Razlika je jedino u tome što zaposlenik ne može uređivati smjene već ih samo označiti.



Slika 3.8 Prikaz sučelja za korisničko popunjavanje rasporeda

Zadatak zaposlenika je odabrati smjene koje želi raditi odabirom kartica smjena. Klikom na karticu smjene, prikazuje se narančasti obrub te se dodaje kartica smjene u sažetak odabira smjena. Po završetku odabira, korisnik pritiskom na gumb „Spremi“ sprema svoj odabir u bazu podataka. Nakon što svi zaposlenici obave ovaj zadatak, slijedi posljednji korak generiranja rasporeda rada.

Imajući odgovore svih zaposlenika, voditelj mora finalizirati raspored. Odabirom rasporeda iz liste prikazuje se tablični prikaz smjena i kartica za definiranje zaposlenika u smjeni. Lista

dostupnih zaposlenika za smjenu se dinamički popunjava promjenom odabira smjene. Voditelj iz liste zatim bira zaposlenike koji će raditi u toj smjeni. Voditelj mora ručno obraditi samo one smjene koje imaju više prijava nego što je potrebno zaposlenika. Ostale smjene se zadano popune svim dostupnim zaposlenicima. Nakon svih odabira, novi se raspored sprema u bazu i zaposlenici koji rade u tom rasporedu dobivaju obavijest putem e-pošte s popisom smjena u rasporedu.

Nakon što je raspored kreiran, također se nudi i opcija generiranja tjednog radnog vremena kluba u .pdf formatu po dogovorenom predlošku. Dužnost svakog poslovnog objekta je imati jasno izloženo radno vrijeme pa tako i noćnih klubova. Generiranjem .pdf dokumenta na temelju već spremljenog rasporeda se u potpunosti automatizirao zadatak koji se dosada obavljao ručno.

### 3.4.5. Radni sati i plaće

Radni sati zaposlenika se automatski računaju iz rasporeda i smjena koje su pohranjene. Dva su uobičajena tipa smjene koji se razlikuju po satnici: jutarnji i večernji tip. Postavljen je i treći tip generičkog naziva „Ostalo“ za smjene poput privatnih zabava, iznimno kratke smjene i slično.

Prije isplate, voditelj dobiva na uvid prikaz plaća kako bi ih provjerio i potvrdio. Na te iznose može dodati i dodatne sate ukoliko je to potrebno. Nakon što voditelj potvrdi sve iznose, plaće se spremaju u bazu. Korisnici u svakom trenu mogu vidjeti „zarađeni“ iznos u nekom trenutku izračunat po broju odrađenih smjena. Sučelje za prikaz plaća zaposlenika je prikazano na slici Slika 3.9.

**Plaće**

Mjesečni izračun plaće po terminima Siječanj ▾ 2023. ▾

Satnica za jutarnji: 5,40 €/sat 🕒 TERMINI

Satnica za večernji: 6,00 €/sat

Broj jutarnjih termina: 3 Broj večernjih termina: 2 Broj extra sati (6 €/sat): 0

	16.2		16.2	
Izračun:	$\frac{5,4 \times 3}{16.2}$	$\frac{6 \times 2}{12}$	$\frac{6 \times 0}{0}$	12
			+ 0	28.2

❗ PRIJAVI NEPRAVILNOSTI

Slika 3.9 Prikaz sučelja izračuna plaće zaposlenika

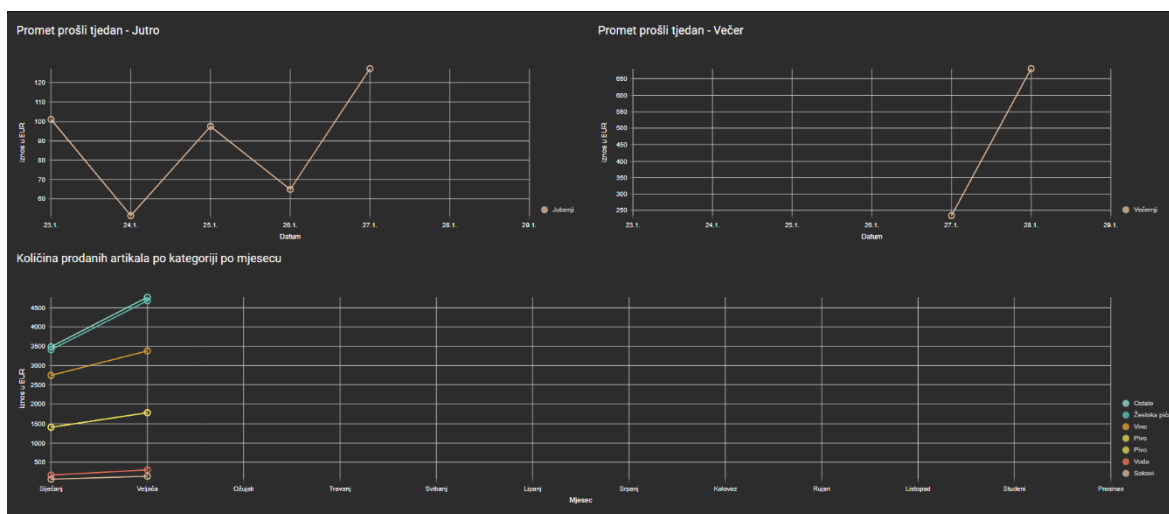
### 3.4.6. Evidencija prometa

Pod evidencijom prometa omogućen je lakši prikaz trenutnog stanja blagajne te praćenje prihoda i rashoda. Svaki promet blagajne se više puta prebrojava i evidentira. Unos prometa se obavlja ručno i ovlasti za ovu operaciju ima samo voditelj. Ovaj način unosa je otprilike ostao isti kao u starom rješenju, ali je novina što se sada ti podaci mogu lako grafički vizualizirati.

### 3.4.7. Grafički prikaz podataka

Grafički prikaz podataka iako nije nužan, uvelike olakšava uočavanje potencijalnih nepravilnosti i grešaka u poslovanju, bilo to stanja skladišta ili prometa ili nešto treće. Grafikoni se koriste za prikaz podataka jer je lakše uvidjeti trendove u podacima kad su vizualno prikazani u usporedbi s brojčanim prikazom u tablici. (Bartee, Shriner, Creech, 2017.)

Na slici Slika 3.10 su prikazani neki grafikoni koji odgovaraju stvarnim podacima generiranim prilikom korištenja aplikacije.



Slika 3.10 Primjer generiranih grafičkih prikaza stvarnim podacima aplikacije

## 4. Analiza rezultata implementacije

Nakon uspješne implementacije i puštanja nove aplikacije u rad u stvarnom poslovnom sustavu, dobiven je uvid u poboljšanja poslovanja i upravljanja opisanih nadalje.

### 4.1. Cilj i metodologija

Nova aplikacija je primarno za cilj imala zadržati i poboljšati funkcionalnosti postojećih korištenih aplikacija, posebice onih koje su navedene kao bitne, poput jednostavnosti korištenja i brzine aplikacije, učitavanja i responzivnosti. Aplikacija je uz to trebala ponuditi poboljšanja rješenja pohrane i integriranosti podataka, lakoće unosa i intuitivnosti, kao i poboljšanje samog dizajna. Ova aplikacija se pokazala korisnom i optimiziranom za navedene zahtjeve poput praćenja duga, obavještanja o rasporedu i stanju skladišta te su korisnici dali dobre ocjene nakon testnog perioda.

Metodologija analize kvalitete novog rješenja je, radi lakoće i efikasnosti usporedbe rješenja, ekvivalentna metodologiji u inicijalnoj analizi. Tako je nakon implementacije i određenog vremenskog korištenja aplikacije provedena ista anketa nad istim skupom korisnika u promatranom primjeru noćnog kluba. Iz inicijalne analize su izbačena sva pitanja osim pitanja treće skupine, tj. pitanja ocjene trenutnog rješenja brojkama od 1 do 10.

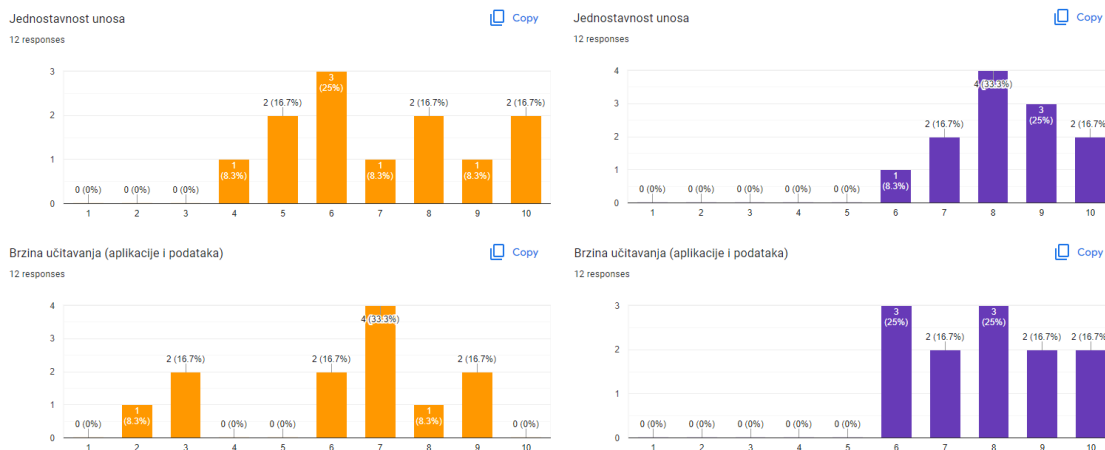
### 4.2. Rezultat analize

Aplikacija je u odnosu na prijašnja rješenja aplikacijama Google Sheets i Google Forms pokazala bolje rezultate i korisničko iskustvo. Prije svega, prosječno vrijeme korištenja aplikacija za ispunjavanje zahtjeva u poslovanju na kraju smjene se smanjilo, sa prosječnih trideset minuta na nešto više od deset minuta.

To uvelike poboljšava korisničko iskustvo, kao i zadovoljstvo zaposlenika koji mora obavljati taj posao svakodnevno. Zadovoljstvo zaposlenika na radu je skup pozitivnih ili negativnih osjećaja koje zaposlenici gaje prema svom poslu (Davis, Newstrom, 1989). Vodeći se tom definicijom, ugodnije iskustvo korištenja aplikacija u radu povećava zadovoljstvo zaposlenika.

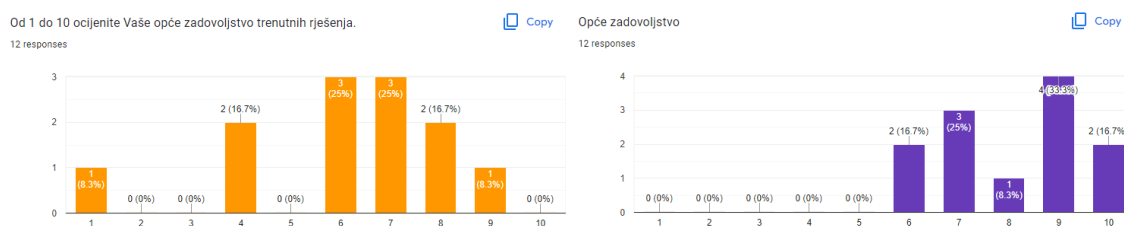
Rezultati analize su pokazali poboljšanje i u ostalim ispitanim karakteristikama. Najbitnija karakteristika iz inicijalne ankete, jednostavnost unosa, je porasla sa prosječne ocjene 6 na

8. Najveći porast se vidi u ocjenama dizajna i prilagodbe malim ekranima. U inicijalnoj anketi je ova karakteristika dobila daleko najlošije ocjene. Novo rješenje je za dizajn i prilagodbu malim ekranima dobila vrlo pozitivne ocjene, u prosjeku 8, a najniža ocjena 5. Poboljšanje se može lako uočiti na grafičkim prikazima raspodjele odgovora anketiranih zaposlenika na slici Slika 4.1.



Slika 4.1 Usporedba inicijalnih rezultata ankete i rezultata druge ankete za značajke jednostavnosti unosa i brzine učitavanja

Ocjene korisničke prilagodbe izgleda i načina te responzivnosti novog rješenja su u prosjeku više, ali znatno manje od već navedenih karakteristika. Bez obzira na malen porast u ocjeni, pokazalo se da je novo rješenje u nekim aspektima postigla barem jednako dobre rezultate kao i staro rješenje. Vidi se također i porast u ocjeni općeg zadovoljstva korištenja aplikacije koje je prikazano na slici .



Slika 4.2 Usporedba odgovora zaposlenika na opće zadovoljstvo korištenja aplikacije u prvoj i drugoj anketi

Voditelji koji nadgledaju unose i rad zaposlenika u aplikaciji su u prosjeku zadovoljniji, budući da je vidljivo smanjen broj pogrešaka koje su se događale prilikom unosa podataka, bilo to u obračunu ili stanju skladišta na kraju smjene.

Naposlijetku, rezultati su ankete pokazali olakšano upravljanje dinamičkim potrebama poslovanja noćnog kluba, poput upravljanja rasporedom rada zaposlenika i parametrima

korištenih artikala. Preciznije, smanjeno je prosječno vrijeme obavljanja repetitivnih poslova. Povećanjem zadovoljstva radnika smo pripomogli budućem poslovanju noćnog kluba jer radnici, što su zadovoljniji na radu, ostvaruju bolje individualne performanse. (Bakotić, Vojković, 2013.)



## Zaključak

Analiza poslovanja noćnog kluba je pokazala česte dinamične potrebe i specifičnosti organizacije. Specifičnosti poput uzimanja pića na dug i izrade rasporeda rada iz tjedna u tjedan među više od deset zaposlenika i nisu toliko česta pojava. Međutim, javlja se potreba za nekakvim alatom koji će olakšati praćenje i funkcioniranje već postojeće organizacije i procesa.

Zaposlenicima je kod korištenja aplikacija u poslu najbitnija jednostavnost i brzina unosa. Drugim riječima, zaposlenik želi što prije moguće obaviti posao na računalu. Aplikacija ne smije svojim sporim učitavanjem kočiti zaposlenika u poslu. Da bi se ovi i svi ostali zahtjevi ispunili potrebno je napraviti kvalitetnu implementaciju novog rješenja.

Implementacija novog rješenja je izrađena pomoću široko korištenih razvojnih okvira ASP.NET i React.js kako bi odražavanje u budućnosti bilo jednostavnije. Zahtjevi korisnika su uzeti u obzir pri izradi aplikacija, pa tako i voditelja za upravljačko sučelje koje broji mnogo više funkcionalnosti dostupnih običnim zaposlenicima. Konzistentnost dizajna i jednostavnost unosa su bili u fokusu implementacije.

O uspješnosti implementacije novog rješenja govori završna analiza implementacije. Korisnici, točnije zaposlenici, su kvalitetnije ocijenili novo rješenje u usporedbi sa starim. Iz tih boljih ocjena možemo zaključiti da su potrebe kluba dobro uočene te na kvalitetan način izvedene. Novo rješenje nije zatvoreno i jednostavno ga je proširiti novim značajkama i funkcionalnostima poput mobilne inačice aplikacije i funkcionalnosti fiskalne blagajne.

## Popis kratica

SPA *Single-page app*

SEO *Search Engine Optimization*

SSR *Server-Side Rendering*

Jednostrana aplikacija

Optimizacija za tražilice

Prikaz na poslužiteljskoj strani

## Popis slika

Slika 2.1 Grafički prikaz odgovora ankete za dvije ispitanicima najbitnije karakteristike ...	4
Slika 3.1 Sekvencijalni dijagram zahtjeva klijentske aplikacije prema bazi podataka .....	9
Slika 3.2 Primjer rezpozivnosti dizajna aplikacije u odnosu na širinu zaslona u pikselima .....	11
Slika 3.3 Stranica za prijavu korisnika u sustav .....	12
Slika 3.4 Prikaz unosa podataka pri inventuri .....	14
Slika 3.5 Prikaz kreiranja lozinke pri registraciji novog korisnika .....	15
Slika 3.6 Prikaz modalnog prozora za brzo dodavanje tj. oduzimanje iznosa posudbe .....	16
Slika 3.7 Prikaz sučelja za izradu tjednog rasporeda.....	16
Slika 3.8 Prikaz sučelja za korisničko popunjavanje rasporeda .....	17
Slika 3.9 Prikaz sučelja izračuna plaće zaposlenika.....	18
Slika 3.10 Primjer generiranih grafičkih prikaza stvarnim podacima aplikacije .....	19
Slika 4.1 Usporedba inicijalnih rezultata ankete i rezultata druge ankete za značajke jednostavnosti unosa i brzine učitavanja .....	21
Slika 4.2 Usporedba odgovora zaposlenika na opće zadovoljstvo korištenja aplikacije u prvoj i drugoj anketi.....	21

## Literatura

- [1] NET SOLUTIONS, What is Single Page Application (SPA)? Pros and Cons with Examples, <https://www.netsolutions.com/insights/single-page-application/>, 2017., preuzeto u siječnju 2023.
- [2] LOCK, A. *ASP.NET Core in Action, Second Edition*. Manning, 2021.
- [3] BARTEE, L., SHRINER, W., CREECH C. *Principles of Biology*, <https://openoregon.pressbooks.pub/mhccmajorsbio/chapter/presenting-data>, siječanj 2023.
- [4] COYLE A., The Responsive Table, <https://coyleandrew.medium.com/the-responsive-table-44d533fde1e9>, 2014., preuzeto u siječnju 2023.
- [5] DAVIS, K., NEWSTORM, J. W., *Human Behavior at Work*, McGraw-Hill, 1989., str. 176.
- [6] STATISTA.COM, Most used web frameworks among developers worldwide, as of 2022, <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>, 2022., preuzeto u siječnju 2023.
- [7] BAKOTIĆ, D., VOJKOVIĆ, I. Veza između zadovoljstva na radu i individualnih performansi. *Poslovna izvrsnost: znanstveni časopis za promicanje kulture kvalitete i poslovne izvrsnosti*, 7, 1, 2013., str. 40

# Prilog

## Anketa o trenutnim aplikativnim rješenjima korištenih u poslovanju

### Opća pitanja

- Koje zadatke u svom opisu posla rješavate aplikacijama Google Sheets i Google Forms?
- Koje su prednosti rješavanja poslova, navedenih u prošlom odgovoru, uz pomoć aplikacija Google Sheets i Google Forms?
- Koliko vremena potrošite na korištenje aplikacija u jednoj smjeni? Svoju procjenu napišite u minutama.

**Važnost karakteristika aplikacije** – odgovore unesite brojkom od 1 do 10 gdje 1 predstavlja „nebitno“, a 10 „vrlo bitno“.

- Brzina učitavanja (aplikacije i podataka)
- Jednostavnost unosa
- Korisnička prilagodba izgleda i načina unosa (npr. veličina fonta, unos putem tipkovnice/tipki na ekranu...)
- Responzivnost aplikacije (brzina kojom aplikacija "odgovori" na korisničku akciju)
- Dizajn i prilagodba malim ekranima

**Ocjena karakteristika trenutnih rješenja** – odgovore unesite brojkom od 1 do 10 gdje 1 predstavlja „loše“, a 10 „odlično“.

- Brzina učitavanja (aplikacije i podataka)
- Jednostavnost unosa
- Korisnička prilagodba izgleda i načina unosa (npr. veličina fonta, unos putem tipkovnice/tipki na ekranu...)
- Responzivnost aplikacije (brzina kojom aplikacija "odgovori" na korisničku akciju)
- Dizajn i prilagodba malim ekranima

### Osobni dojam trenutnih aplikativnih rješenja

- Prednosti trenutnih rješenja
- Mane trenutnih rješenja
- Značajke koje mislite da nedostaju, a htjeli biste ih imati?
- Koje su najčešće greške koje se javljaju pri radu s aplikacijama?
- Od 1 do 10 ocijenite Vaše opće zadovoljstvo trenutnih rješenja.

### Završna pitanja

- Koji ostali zadaci koje obavljate u svom poslu mogu biti olakšani aplikativnim rješenjem? U odgovoru navedite svoj prijedlog kako bi rješenje funkcioniralo.
- Ostale napomene koje smatrate da treba uzeti u obzir pri izradi aplikacije.