

Pregled primjenjivih oblika monetizacije odabranog tipa programskih proizvoda

Krmpotić, Andrija

Master's thesis / Diplomski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:366465>

Rights / Prava: [Attribution-NoDerivs 3.0 Unported/Imenovanje-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2025-03-20**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Andrija Krmpotić

**PREGLED PRIMJENJIVIH OBLIKA
MONETIZACIJE ODABRANOG TIPA
PROGRAMSKIH PROIZVODA**

DIPLOMSKI RAD

Varaždin, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Andrija Krmpotić

Matični broj: 0016127095

Studij: Baze podataka i baze znanja

**PREGLED PRIMJENJIVIH OBLIKA MONETIZACIJE ODABRANOG
TIPA PROGRAMSKIH PROIZVODA**

DIPLOMSKI RAD

Mentorica:

Prof. dr. sc. Katarina Tomičić-Pupek

Varaždin, veljača 2025.

Andrija Krmpotić

Izjava o izvornosti

Izjavljujem da je ovaj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI Radovi

Sažetak

U diplomskom radu napravila bi se detaljna analiza ključnih modela monetizacije programskih proizvoda odabranog tipa. Svaki model dolazio bi uz objašnjen primjer koji bi jasno prikazao oblik monetizacije koji se koristi za određeni programski proizvod. Glavni primjer koji kombinira niz različitih oblika monetizacije realizirao bi se na detaljnoj analizi popularne stranice za filmove i serije, IMDb (Internet Movie Database). Kao praktični dio rada bit će napravljen prototip aplikacije u programskom jeziku Blazor i .Net razvojnom okviru koji će uklopiti niz odabranih modela monetizacije.

Ključne riječi: monetizacija; programski proizvod; blazor; IMDb; MAUI; mobilna aplikacija

Sadržaj

1. Uvod	1
2. Propozicije vrijednosti	2
2.1. Privatnost i anonimnost	2
2.2. Averzija gubitka	2
2.3. Status i ugled	3
2.4. Misteriozne kutije	3
2.4.1. Bit kupnje	4
2.4.2. Procedura izvlačenja	4
2.4.3. Sadržaj	5
2.4.4. Audiovizualni aspekt	5
2.4.5. Neplaćeni angažman	5
2.4.6. Socijalni aspekt	5
2.5. Ekskluzivnost i ograničena dostupnost	6
3. Treća strana plaća	8
3.1. Oglašavanje	8
3.1.1. Oglasi u obliku transparenta	8
3.1.2. Prirodni oglasi	8
3.1.3. Video oglasi	9
3.2. Tržište unutar aplikacije	9
3.3. Prodaja podataka	10
3.3.1. Zakonske regulative	10
3.3.2. Razlike između dijeljenja i prodaje podataka	11
4. Besplatne opcije za korisnike	12
4.1. Donacije	12
4.2. Plati koliko želiš	12
4.3. Freemium	13
5. Cjenovne strategije	15
5.1. Pretplata	15
5.2. Plaćanje po korištenju	15
5.3. Dinamično određivanje cijena	16
5.4. Jednokratna naknada	17
5.5. Pretprodaja	17

5.6. RazorBlade model	18
6. Analiza primijenjenih oblika monetizacije na primjeru Internet Movie Database programskog proizvoda	20
6.1. Oglasi	20
6.2. IMDbPro Pretplate	20
6.3. Partnerstva i Licenciranje	21
6.4. Prodaja putem IMDb Trgovine	21
6.5. Amazon Podružnički Program	21
7. Praktični dio	22
7.1. Baza podataka	22
7.1.1. Kod baze	24
7.2. Stranice aplikacije	25
7.2.1. Prijava	25
7.2.2. Poslovi	27
7.2.3. Kreiranje oglasa za posao	30
7.2.4. Profil korisnika	31
7.3. Primijenjene tehnike monetizacije aplikacije	33
8. Zaključak	35
Popis literature	37
Popis slika	38
Popis isječaka koda	39

1. Uvod

Razvoj softverske industrije u posljednja dva desetljeća doveo je do značajnog porasta popularnosti modela "Programskih proizvoda u obliku usluge" (*eng. Software as a Service*) (U daljnjem tekstu: SaaS), koji je radikalno promijenio način na koji poduzeća i pojedinci koriste softverske aplikacije. SaaS omogućava korisnicima pristup programskim proizvodima putem interneta na temelju pretplate, eliminirajući potrebu za instalacijom i održavanjem aplikacija na lokalnim uređajima. Ova transformacija je olakšala pristup programskim proizvodima, smanjila troškove i povećala fleksibilnost za korisnike, što je dovelo do ekspanzije ovog modela diljem industrije.

Monetizacija u SaaS-u odnosi se na strategije i metode koje tvrtke koriste kako bi generirale prihod od svojih programskih rješenja. Dok tradicionalni modeli prodaje softvera obično uključuju jednokratne kupnje, SaaS omogućava kontinuirano generiranje prihoda putem pretplata i drugih modela plaćanja. Ove tehnike uključuju pretplate, freemium modele, kao i ekskluzivnost i ograničenu dostupnost. Svaka od ovih metoda nosi svoje prednosti i izazove, te je od iznimne važnosti razumjeti ih kako bi se postigao optimalan financijski rezultat.

Monetizacija u SaaS industriji je dinamičan proces koji zahtijeva pažljivo planiranje i prilagodbu različitim tržišnim segmentima. Tvrtke moraju balansirati između pružanja vrijednosti korisnicima i maksimizacije prihoda, što zahtijeva duboko razumijevanje tržišnih potreba i preferencija korisnika. Uz to, važnost inovacija i kontinuiranog unaprjeđenja proizvoda ne smije se zanemariti, jer konkurencija u SaaS prostoru stalno raste. Ovaj rad ima za cilj pružiti sveobuhvatan pregled različitih monetizacijskih tehnika u SaaS industriji, analizirati njihove prednosti i izazove, te ponuditi konkretne primjere uspješnih implementacija iz prakse.

2. Propozicije vrijednosti

Propozicije vrijednosti (*engl. Value proposition*) su strateški alat koji olakšava komunikaciju o sposobnosti organizacije da dijeli resurse i ponudi superiorni paket vrijednosti ciljanim kupcima, stoga je bitan aspekt prilikom monetizacije programskog proizvoda. [1] Potencijalnim kupcima programskog proizvoda bitno je prikazati vrijednosti koje on nudi kako bi se kupce zaintrigiralo za navedeni proizvod, a time i povećala prodaja i popularnost samog proizvoda. U sljedećim dijelovima rada detaljnije su objašnjeni mogući koncepti predstavljanja vrijednosti potencijalnim kupcima proizvoda.

2.1. Privatnost i anonimnost

Privatnost i anonimnost su dvije vrlo atraktivne opcije klijentima koji cijene sigurnost i zaštitu svojih podataka. Klijenti su spremni platiti više kako bi dodatno osigurali sve njihove podatke te ostali anonimni pri korištenju programskih proizvoda ili internetskih preglednika. Jednostavan primjer može se naći u mnogobrojnim programskim proizvodima koje se skupno nazivaju virtualne privatne mreže (u daljnjem tekstu VPN). Ovakvi programi klijentu nude da šifriraju svoj internetski promet i sakriju svoju IP adresu prilikom surfanja internetom. Ovakvi programi često se naplaćuju prema potrošenom prometu ili u slučaju tvrtke, prema korisniku, odnosno uređaju koji koristi program.

U okviru plaćenih programa za programski proizvod korisniku se može ponuditi opcija kreiranja anonimnih računa u kojima se ne bilježi nikakva korisnička aktivnost i njihova identifikacija ostaje skrivena drugim korisnicima. Isto tako premium korisnicima može se ponuditi opcija samostalnog upravljanja podacima što daje dodatnu kontrolu i dodaje sigurnosni sloj nad njihovim podacima. Opcije koje se u tom kontekstu mogu ponuditi korisnicima su lokalno pohranjivanje podataka ili samostalno upravljanje ključevima šifriranja što u potpunosti ograničava pristup podacima isključivo tim korisnicima. [2]

Kako bi se osiguralo povjerenje klijenta, poduzeće mora transparentno komunicirati vrijednosti koje dodatne funkcionalnosti za privatnost i anonimnost dodaju u proizvod. Poduzeća također nude pravnu zaštitu i savjetovanje kako bi se klijenta dodatno podučilo o sigurnosti njihovih podataka, a samim time i pridobilo povjerenje klijenta. Bitno je spomenuti kako cijena mora odražavati dodatne troškove i trud koji su potrebni za implementaciju navedenih mjera, a klijenti moraju biti uvjereni da su njihovi podatci zaista sigurni.

2.2. Averzija gubitka

Korisnički podatci vrijedni su davateljima usluge, ali i korisnicima same usluge. Korisnici su često spremni platiti više kako bi ti podatci bili dostupni njima na korištenje nakon nepogodnog događaja kao što su napadi hakiranja, pad napona, uništenje lokalne pohrane podataka i slično. Ako se gubitak podataka već dogodio tada je u interesu korisnika da može povratiti izgubljene podatke koji su pohranjeni na sigurnom mjestu. No, averzija gubitka nije samo u ču-

vanju podataka. U ovu kategoriju mogu se uključiti i pravna savjetovanja, korisnička podrška, odnosi s javnošću i sl. Primjer jednog takvog programskog proizvoda može se pronaći u poduzeću Salesforce pod nazivom "*Salesforce Shield*" koji uključuje funkcionalnosti poput enkripcije podataka, naprednih opcija praćenja i pohranjivanja podataka i sigurnosnih kopija. Amazon Web Services (AWS) nudi programske proizvode "*AWS Shield*" [3] i "*AWS Guard Duty*" [4] koji služe za otkrivanje distribuiranih napada uskraćivanja usluge (*eng. Distributed denial-of-service attack - DDoS*) te ostalih sigurnosnih prijetnji u sustavu pohrane podataka. Poduzeća koja nude ovakav tip zaštite podataka često imaju posebne pakete za velika poduzeća koja se naplaćuju ovisno o broju funkcionalnosti koje poduzeće ugovori i njihovoj opsežnosti.

2.3. Status i ugled

Prilikom odabira programskog proizvoda potencijalni kupac također može gledati status i ugled koji će mu taj programski proizvod donijeti. Čak i samo postojanje sustava za rangiranje unutar samog programskog proizvoda dovoljno je da pridonese dopadljivosti proizvoda kupcima. Primjer boljeg statusa unutar zajednice koja koristi programski proizvod su najčešće "premium" korisnici koji imaju dodatne pogodnosti te je samim time i njihov status uzdignut stepenicu iznad ostalih korisnika. Prilikom korištenja programskih proizvoda koji pružaju samo usluge to se može jasno vidjeti kod platformi koje imaju inicijativu samozapošljavanja na određenim zadacima. Primjeri takvih platformi su: "Fiverr", "PieceX", "HireBasis", "Upwork" i sl. Takve platforme svojim korisnicima omogućuju da svoj status i reputaciju pokažu putem broja uspješno obavljenih zadataka koji su potražitelj javno postavlja, a sam koji isto tako može ocijeniti samu izvedbu zadatka.

Mogućnost prikazivanja statusa i reputacije unutar programskog proizvoda na specifičan način potiče korisnike da potroše stvarni novac kako bi ostvarili taj status. Ovakav način trgovanja omogućuje visoku zaradu putem mikrotransakcija unutar sustava. [5]

2.4. Misteriozne kutije

Misteriozne kutije (*engl. Mystery Box*) u današnje vrijeme popularne su kod videoigara, ali je koncept primjenjiv na većinu programskih proizvoda. Kako bi koncept monetizacije na temelju misterioznih kutija bio uspješno implementiran u programski proizvod bitno je kreirati strategiju implementacije koja će pokriti sljedećih šest kategorija[6]:

- Bit kupnje - Zbog čega će korisnici kupiti kutiju?
- Procedura izvlačenja - Način odabira nagrade unutar kutije.
- Sadržaj - Kojeg oblika je sadržaj unutar kutije i što donosi kupcu.
- Audiovizualni aspekt - Psihološki aspekt nagrade popraćen zvukovima i žarkim bojama.
- Neplaćeni angažman - Besplatni uzorci kutija kroz vrijeme.

- Društvena vrijednost - Vidljivost sadržaja unutar programskog proizvoda, mogućnost razmjene sadržaja s drugim korisnicima.

2.4.1. Bit kupnje

Kada se gleda bit kupnje misteriozne kutije bitno je spomenuti pet aspekata koji direktno utječu na proces kupnje misteriozne kutije. Prvi aspekt je valuta kojom se kupuje. Valuta može biti realan novac koji se koristi u stvarnom svijetu, no može biti i virtualna kao na primjer novac koji se skuplja unutar igre. Drugi aspekt je izravnost kupnje. Izravnost kupnje može se podijeliti na četiri potkategorije:

- Kutije se kupuju direktno.
- Kutije se dobiju putem kupnje određenih paketa.
- Kupuju se određeni artikli koji generiraju kutije, ali se kutije ne kupuju direktno.
- Kutije se dobiju putem izvršavanja određenih zadataka unutar programskog proizvoda.

Treći aspekt karakterizira se u dostupnosti misterioznih kutija te se može podijeliti na kupovanje kutija izvan programskog proizvoda putem drugih servisa prodaje ili direktno unutar programskog proizvoda. Četvrti aspekt je prezentiranje postotaka za određenu razinu vrijednosti. U videoigrama se često pokazuju šanse za dobivanje rjeđih artikala kako bi se psihološki utjecalo na igrača u korist kupnje kutije, no postoje i slučajevi gdje su postotci potpuno skriveni te igrač ne zna kakve šanse ima za ostvarivanje rijetkih artikala. Peti aspekt su popusti s ograničenim vremenskim trajanjem. Ovdje se mogu ponuditi kutije s većim šansama za boljim artiklima na ograničeno vrijeme ili je moguće na određeno vrijeme postaviti popuste na kutije kako bi se igrač natjeralo da kupuju više kutija.[6]

2.4.2. Procedura izvlačenja

Procedura izvlačenja sadržaja zapravo je sustav koji određuje koji sadržaj će korisnik dobiti prilikom otvaranja misteriozne kutije. Kod procedure izvlačenja sadržaja iz kutije bitno je spomenuti tri ključne stvari: opskrba artiklima (*Item supply*), rukovanje duplikatima (*Duplicate handling*) i rukovanje neželjenim sadržajem. Prilikom opskrbe artiklima bitno je razmisliti koji sadržaj je moguće izvući iz misteriozne kutije, a koji je potrebno kupiti zasebno. Ključno pitanje koje se mora postaviti prilikom odabira razine rariteta sadržaja unutar misteriozne kutije je: "U kojim postotcima će se pojavljivati određene razine rariteta sadržaja?". U slučaju da određeni sadržaj ima premalu ili preveliku šansu izvlačenja iz kutije, on gubi na vrijednosti. Stoga je vrlo bitno precizno odrediti razine rariteta sadržaja. Rukovanje duplikatima može se riješiti na više načina. Duplikate je moguće pretvoriti u virtualnu valutu s kojom je moguće kupiti još artikala ili potencijalno omogućiti korisniku da pretvori sadržaj kutije u nešto drugo, ali garantirane

manje vrijednosti nego prethodni artikl. Neželjeni sadržaj često je onaj sadržaj najlošije kvalitete u misterioznim kutijama te je njega najviše, stoga je bitno ponuditi prije spomenutu opciju zamjene sadržaja za određenu vrijednost. [6]

2.4.3. Sadržaj

Sadržaj kutije često je karakteriziran nekim oblikom vrijednosti. Ta vrijednost može se percipirati na različite načine no bitno je specificirati razrede ili klase vrijednosti. Primjer klase vrijednosti artikla, običan (*engl. common*), rijedak (*engl. rare*), vrlo rijedak (*engl. very rare*), legendaran. Isto tako sadržaj je moguće podijeliti i na sezonski i nesezonski sadržaj te njegovu vrijednost postaviti tako da se isključivo može dobiti putem misterioznih kutija. [6]

2.4.4. Audiovizualni aspekt

Kako bi kupac kutije bio zaintrigiran misterioznom kutijom, potrebno ju je prezentirati u najboljem mogućem izdanju. Takvo izdanje uključuje žarke, privlačne boje same kutije. Animacije prilikom otvaranja kutije pridonose generiranju dobrog osjećaja kod kupca. Popratni zvukovi koji se puštaju prilikom kupnje i otvaranja kutije potencijalno mogu stvoriti ovisnički efekt koji kupca navodi na kupnju dodatnih kutija ili proizvoda.

Primjer dobre implementacije je animacija kada korisnik zamalo dobije (*engl. Near miss*) najvrjedniji sadržaj kutije, odnosno pokazivanje onoga što je mogao osvojiti. Na taj način, korisnik je motiviran za ponovnu kupnju kutije.[6]

2.4.5. Neplaćeni angažman

Neplaćeni angažman zapravo opisuje načine na koje je moguće ohrabriti kupce da stupe u interakciju sa sustavom misterioznih kutija. Dva su moguća načina interesiranja potencijalnih kupaca za kupnju, a to su:

- Besplatni uzorci
- Promotivne ponude

Kod besplatnih uzoraka, koji se najčešće nude prilikom završetka određenih zadataka koje sustav postavlja, mogu se direktno ponuditi misteriozne kutije ili je kupcu moguće ponuditi internu valutu unutar sustava s kojom se kupuju misteriozne kutije. Promotivne ponude najčešće dolaze uz ograničeno vrijeme. Moguće je kupiti jednu ili više misterioznih kutija po promotivnoj cijeni uz plaćanje s internom *premium* valutom ili stvarnom valutom. [6]

2.4.6. Socijalni aspekt

Najbitnija stavka unutar socijalnog aspekta jest vidljivost ili utjecaj koji je sadržaj iz kutije donio kupcu. Ova stavka može se realizirati u obliku modifikacije korisničkog profila, korištenjem pogodnosti pa čak i promjenama tema boja u sustavu. Postoji puno promjena koje mogu

utjecati na vidljivost, ali sve promjene svode se na mogućnost drugih korisnika da primijete promjenu kod originalnog kupca kutije. [6]

2.5. Ekskluzivnost i ograničena dostupnost

Ekskluzivnost unutar SaaS (*eng. Software as a Service*) industrije odnosi se na strategiju kojom se određeni softverski proizvodi ili usluge čine dostupnima pod posebnim uvjetima ili samo određenom broju korisnika. Korištenje ove strategije može uključivati visoku cijenu, ograničen pristup, specifične zahtjeve za ulazak, personalizirani pristup, limitirane resurse, ekskluzivni sadržaj ili kombinacije navedenih faktora. Primjer jedne takve aplikacije koja pruža uslugu slanja elektroničke pošte naziva se "Superhuman"[7] te kombinira sve faktore za kreiranje ekskluzivnog ambijenta unutar programskog proizvoda. Ako osoba želi postati korisnik ove aplikacije, potrebna joj je pozivnica kako bi se priključila bazi korisnika. Pozivnice se mogu dobiti od strane postojećih korisnika ili se osoba može prijaviti na listu čekanja kako bi kroz određeni period primila pozivnicu za pridruživanje u aplikaciju. Lista čekanja pridodaje dojamu ekskluzivnosti i povećava osjećaj iščekivanja i vrijednosti usluge. Drugi faktor koji igra veliku ulogu u dojamu ekskluzivnosti programskog proizvoda jest visoka cijena. Visoka cijena postiže efekt ozbiljnosti i profesionalnosti te je namijenjena korisnicima koji su spremni platiti više kako bi dobili kvalitetnu uslugu. Prilikom dugo iščekivanog pristupa aplikaciji, korisnik od strane stručnjaka aplikacije dobiva individualni proces uvođenja. Ovaj postupak uvelike podiže osjećaj ekskluzivnosti i pripadnosti "elitnoj" grupi korisnika koji koriste ovu uslugu. [8]

Kada se govori o ekskluzivnosti unutar SaaS okvira najčešće se stavlja fokus na prednosti korištenja ove strategije. Glavni cilj, koji je ujedno i prednost, je povećanje vrijednosti usluge. Korisnici ekskluzivnih usluga često te usluge percipiraju kao vrijednije te samim time sebi opravdavaju visoke cijene usluge i dulje vremena ostaju vjerniji jednom ekskluzivnom programskom proizvodu. Bitan aspekt održavanja ekskluzivnosti je kontrola kvalitete usluge. S manjim brojem korisnika lakše je osigurati visoku razinu kvalitete i bolje upravljati korisničkim iskustvom prilikom unaprjeđenja ili modificiranja usluge. Još jedna prednost koja dolazi s kvalitetom usluge je povećana potražnja. Ekskluzivnost može stvoriti osjećaj hitnosti i želje za posjedovanjem navedenog programskog alata što dovodi do povećanog interesa i potražnje same usluge. Isto tako bitno je navesti mogućnost diferencijacije na tržištu usluga. SaaS tvrtke s ekskluzivnim uslugama lakše se izdvajaju od konkurencije te privlače specifične segmente tržišta ako imaju precizno definiranu ciljanu publiku kojoj je potrebna usluga koju nude.

Korištenje ekskluzivnosti u SaaS modelu može imati i negativne posljedice koje treba posebno razmotriti prije implementacije same strategije. Na primjer, ograničavanjem pristupa usluzi može biti posebno problematično za manje tvrtke kojima je potreban rast i velik broj korisnika kako bi se probile na tržište. Ograničavanje pristupa pridonosi efektu "elitizma" koji odbija širu publiku koja bi potencijalno mogla biti zainteresirana za uslugu. Za manji broj korisnika koji zahtijevaju visoku razinu personalizirane podrške ponekad je potrebno organizirati cijele timove kako bi razina kvalitete ostala visoka. Odražavanje kvalitete s brojem korisnika koji se konstantno povećava prilično je izazovno i financijski i vremenski te stoga predstavlja rizik nemogućnosti skaliranja usluge, a prestanak rasta i nemogućnost dovođenja novih korisnika

često dovodi do smanjenja profita.

Ekskluzivnost kao strategija može biti izuzetno učinkovita u SaaS industriji, posebno kada je usmjerena na specifične segmente tržišta koje cijene personaliziranu i kvalitetnu uslugu. Isto tako bitno je pažljivo procijeniti moguće negativne strane i osigurati da strategija ekskluzivnosti ne ugrožava dugoročne ciljeve rasta i zadovoljstva korisnika. [9]

3. Treća strana plaća

Treća strana (*eng. Third party*) može se definirati kao entitet koji nije direktno uključen u transakciju ili odnos između dva primarna entiteta. U tehnološkom svijetu, treća strana odnosi se na programera, tvrtku ili pružatelja usluga koji nije dio originalnog tima za razvoj programskog proizvoda, ali nudi dodatke, proširenja ili dodatne usluge unutar primarnog programskog proizvoda. U svim slučajevima treća strana karakterizira se kao entitet koji nije direktno uključen u glavni odnos, ali može imati ulogu ili biti pod utjecajem postupaka glavnih strana.

3.1. Oglašavanje

Oglašavanje unutar programskih proizvoda danas je najrašireniji pristup priljeva novca od strane poduzeća kojima je potrebna reklama. U današnje vrijeme gotovo u svakoj se besplatnoj aplikaciji i web stranici nalaze reklamni prostori koji se prodaju po određenim cijenama. Reklamni sadržaji unutar SaaS modela mogu biti raznovrsni, s ciljem maksimizacije prihoda i angažmana korisnika. Evo nekoliko ključnih tipova reklamnih sadržaja koji se obično koriste:

- Oglasi u obliku transparenta (*eng. Banner ads*)
- Prirodni oglasi (*eng. Native ads*)
- Video oglasi

3.1.1. Oglasi u obliku transparenta

Transparent kao način oglašavanja najčešće se koristi na web stranicama ili unutar aplikacija svih vrsta. To je grafička vrsta oglasa koja unutar jedne slike ili infografike sadrži sve informacije koje se žele prenijeti korisniku. Transparenti se uglavnom ističu velikim naslovima i žarkim bojama kako bi privukli pažnju korisnika. Visoka vidljivost primarna je prednost ovakvog tipa oglasa, dok se broj klikova i učinkovitost lako može mjeriti. Još jedna bitna osobina ovakvog tipa oglašavanja je mogućnost ciljanja specifičnih demografskih skupina. Transparenti se najčešće koriste kod raznih web trgovina kada tvrtka želi istaknuti jedan od njihovih proizvoda ili popuste na određenu skupinu proizvoda koja je trenutno aktivna.

3.1.2. Prirodni oglasi

Prirodni ili "*Native*" oglasi su integrirani u sadržaj web stranice ili aplikacije tako da izgledaju kao prirodni dio korisničkog iskustva. Cilj im je da budu manje invazivni i pružaju korisnicima relevantan sadržaj. Ključna karakteristika prirodnih oglasa je da zapravo ne izgledaju kao oglasi nego da se stope sa sadržajem oko njih te ne prekidaju korisnikovu interakciju sa stranicom. Njihova glavna prednost je veća razina interakcije, a sama priroda ovakvih oglasa manje je invazivna od transparentata na stranicama. Isto tako ovaj tip oglasa kreiran je tako da, uz primarnu funkciju prezentiranja proizvoda kupcu, upotpuni korisničko iskustvo pružajući

korisniku oglase koji su bliski sadržaju stranice na kojem se korisnik trenutačno nalazi. Primjeri ovakvog tipa oglasa mogu se očitovati u promotivnim člancima na platformama poput daily.dev platforme čiji je cilj pronalaženje relevantnih članaka na temu iz IT svijeta koju korisnik preferira te uz tu temu može ponuditi i neke članke u obliku prirodnog oglašavanja.

3.1.3. Video oglasi

Video oglasi su kratki videozapisi koji se prikazuju prije, tijekom ili nakon video sadržaja na web stranicama ili unutar aplikacija. Ovi oglasi mogu biti vrlo učinkoviti zbog svoje dinamičke prirode. Mnogi SaaS proizvodi koriste video oglase kao njihov preferirani način oglašavanja. Primjer se može pronaći na YouTube platformi gdje je video oglas postavljen između ostalih videa koje je korisnik odabrao. Ovakav način oglašavanja može biti invazivan, ali zauzima potpunu pažnju korisnika jer najčešće predstavlja jedinu dinamičku komponentu web stranice u vremenu trajanja oglasa. Porast ovakvog oblika oglašavanja vrlo je vidljiv na društvenim mrežama poput Facebooka, Instagrama i Tik-Toka. Na ovim mrežama video oglasi prevladavaju u obliku kratkih videa između ostalih videa korisničkog porijekla pod nazivom "*Reel*" ili "*Tik-Tok*".

3.2. Tržište unutar aplikacije

Kada se govori o tržištu unutar aplikacije mora se postaviti granica između SaaS-a koji je tržište i SaaS-a koji ima ugrađeno tržište unutar svog sustava. Postoje i kombinacije ovih oblika programskih proizvoda. Izvrstan primjer oba oblika može se pronaći u platformi "*Steam*" [10] kojoj je primarna djelatnost tržište videoigara, ali isto tako posjeduje ugrađeno tržište raznim objektima kozmetičke naravi koji se mogu koristiti na samoj platformi ili unutar videoigara koji se kupuju pravim novcem ili valutom specifičnom za navedenu platformu.

Platforma *Steam* povezuje se s proizvođačima videoigara te tim proizvođačima nudi velik broj korisnika koji preko platforme mogu doći do željenih videoigara. Kada korisnik kupi videoigru putem *Steam* platforme, postotak od kupnje videoigre odlazi samoj platformi dok kupac plaća punu cijenu igre. Stoga, korisnik plaća cijenu videoigre koju postavlja proizvođač, ali je proizvođač obavezan platiti određeni dio platformi tako da je u ovom slučaju proizvođač videoigre treća strana koja plaća proviziju umjesto kupca kao troškove marketinga na platformi.

Platforma *Steam* ima i sekundarno tržište na kojem se mogu plasirati i proizvodi krajnjih korisnika. To su najčešće kozmetički efekti poput avatara ili animacija profila, ali i raznih modula za pojedine videoigre. Korisnici mogu kreirati svoj sadržaj putem pomoćnog programskog proizvoda nazvanog "*Steam Workshop*". Svi korisnici koji kreiraju svoj sadržaj putem ovog programskog proizvoda vezani su pravnim sporazumom (*eng. Legal agreement*) čije smjernice moraju poštivati kako bi ga mogli koristiti.

Još jedan primjer tržišta unutar aplikacije može se pronaći na platformi za komuniciranje naziva "*Discord*" [11]. Primarna funkcija platforme je kreiranje mjesta za komunikaciju, bila ona tekstualna, audialna ili u obliku videopoziva. Svi elementi koji su dostupni na tržištu unutar platforme isključivo su kozmetičke naravi za korisnički profil. Bitno je napomenuti kako *Discord*

ne posjeduje unutarnju valutu te se svaki objekt unutar *Discord Shop*-a plaća stvarnim novcem.

3.3. Prodaja podataka

Prilikom korištenja programskih proizvoda, korisnik kreira ili dijeli određene podatke. Ti podatci mogu se svrstati u 3 velike skupine: Osobni podatci, tehnički podatci i podatci o ponašanju korisnika. U osobne podatke spadaju podatci poput imena, email adresa, broj mobitela, broj kreditne kartice, geografska lokacija, demografski podatci i slično. Ti podatci su usko vezani uz osobu, odnosno krajnjeg korisnika, koja koristi programski proizvod. Tehnički podatci koji su vezani za korisnika su sljedeći: tip uređaja na kojem se koristi programski proizvod, operacijski sustav koji taj uređaj koristi, tip web preglednika na kojem se koristi programski proizvod, IP adresa korisnika i sl. Podatci koji su od velike važnosti poduzećima te koji se koriste za razne analitike programskog proizvoda su zapravo podaci o korisničkom ponašanju prilikom korištenja proizvoda, a to su: klikovi na linkove, vrijeme provedeno na stranicama, pojmovi koje je korisnik pretražio, određene kupnje ili transakcije i slično. Takvi podatci se mogu koristiti u analitikama kako bi se razvoj programskog proizvoda preciznije usmjerio na korisničke potrebe ili kombinirati s ostalim tipovima podataka kako bi se kreirale personalizirane marketinške kampanje usmjerene direktno prema ciljanoj skupini korisnika.

3.3.1. Zakonske regulative

Sva poduzeća koja prodaju podatke moraju zadovoljiti regulative koje su postavljene u njihovom području poslovanja. Ako poduzeće posluje na području Europe i prodaje podatke, tada mora zadovoljiti regulative GDPR-a (*eng. General Data Protection Regulation*) [12] prilikom prikupljanja podataka. Zanimljivo je da GDPR regulativa ne uključuje eksplicitna pravila za prodaju podataka, samo za prikupljanje i obradu korisničkih podataka. Na primjer, za prodaju podataka dobivenih na području Kalifornije potrebno je poštovati CPRA (*eng. California Public Records Act*) [13] regulativu u kojoj je jedan od zahtjeva da korisnik uvijek mora imati jasno naznačenu opciju gdje može zabraniti programskom proizvodu da prikuplja, obrađuje ili dijeli njegove podatke. Korisnički podatci se u većini zemalja mogu prodavati ako je sam korisnik dao dopuštenje za obradu i prodaju njegovih podataka. Dopuštenje od korisnika se najčešće traži putem kolačića (*eng. Cookies*) ili politikom privatnosti (*eng. Privacy policy*) koja je javno objavljena unutar programskog proizvoda.

Neki korisnički podatci zahtijevaju dodatnu dozvolu od strane korisnika. To su uglavnom osobni podatci od većeg značaja samom korisniku. Na primjer, podatci o zdravlju, broj socijalnog osiguranja ili podaci o bankovnim transakcijama se ne smiju prikupljati niti prodavati. Poduzećima je vrlo bitno da se ponašaju prema zakonskim regulativama o trgovanju podacima jer su novčane kazne vrlo visoke, a u nekim zakonima, poput Južnoafričkog zakona o zaštiti osobnih podataka (*eng. South Africa's Protection of Personal Information Act (POPIA)*), govori se čak i o zatvorskim kaznama.

3.3.2. Razlike između dijeljenja i prodaje podataka

Nakon obrade podataka, poduzeća biraju hoće li obrađene podatke dijeliti ili prodavati. U oba slučaja poduzeće profitira. Ako poduzeće prodaje podatke ostalim poduzećima ili posrednicima podacima tada je isključiva kompenzacija za te podatke u novčanom obliku, a ako poduzeće dijeli podatke tada ta kompenzacija može biti drugačijeg oblika. Na primjer, podatci se mogu dijeliti ostalim poduzećima kako bi ta poduzeća kreirala personalizirane reklame na programskim proizvodima koji će privući direktniju pažnju korisnika. Takav primjer može se vidjeti kod programskih proizvoda poput društvenih mreža ili platformi strujanja (*eng. Streaming platforms*) poput You-tubea, Spotifya i slično. Dijeljenje podataka može se također koristiti u svrhu poboljšanja same usluge. Na primjer, obrađeni podatci podijele se s konzultantskim poduzećem koje može usmjeriti razvoj programskog proizvoda kako bi ga više približio korisničkim zahtjevima.

4. Besplatne opcije za korisnike

Unutar različitih monetizacijskih modela za SaaS proizvode, besplatne opcije igraju ključnu ulogu u privlačenju korisnika i izgradnji baze korisnika. Freemium model je najčešći, gdje osnovna verzija softvera ili usluge ostaje besplatna, dok napredne značajke zahtijevaju plaćanje. Ovaj model omogućava korisnicima da isprobaju proizvod bez ikakvih troškova, što povećava vjerojatnost da će se odlučiti na nadogradnju ako zadovoljava njihove potrebe. Besplatne opcije ne samo da privlače nove korisnike, već i smanjuju prepreke za ulazak na tržište, omogućujući širenje brenda i povećanje potencijala za konverziju korisnika u plaćene pretplatnike.

4.1. Donacije

Programski proizvod baziran na donacijama je softver ili aplikacija koji se financira kroz dobrovoljne donacije korisnika, umjesto tradicionalnih modela monetizacije poput pretplata, jednokratnih plaćanja ili oglasa. Ovaj model se često koristi za proizvode koji su besplatni za korištenje, gdje korisnici mogu odlučiti podržati daljnji razvoj i održavanje proizvoda putem donacija. Donacije su rijedak oblik monetizacije za SaaS programske proizvode, ali kao primjer može se uzeti navesti aplikacija "*Signal*" [14]. *Signal* je aplikacija za poruke koja svojim korisnicima osigurava sigurnu tekstualnu, audialnu i video komunikaciju bez reklamnih sadržaja i praćenja poruka. Kako navode na svojoj web stranici, aplikacija *Signal* financira se isključivo modelom donacija te odbijaju koristiti ijedan drugi oblik monetizacije.

Kao drugi primjer može se navesti *ProtonMail*. *ProtonMail* je usluga elektroničke pošte koja nije u potpunosti financirana donacijama. *ProtonMail* koristi i ostale oblike poput *freemium* modela i pretplate. Korisnici mogu koristiti osnovne značajke usluge potpuno besplatno, no ako žele koristiti sve pogodnosti potrebno je donirati putem njihove web stranice ili se pretplatiti na uslugu.

Ovi SaaS proizvodi pokazuju da model financiranja putem donacija može biti održiv, osobito kada korisnici prepoznaju društvenu ili osobnu vrijednost u proizvodu koji koriste. Takvi proizvodi često apeliraju na etičku svijest korisnika, kao što je privatnost, otvoreni kod ili slobodan pristup informacijama.

4.2. Plati koliko želiš

"Plati koliko želiš" (*eng. Pay-what-you-want*) [15] je mehanizam određivanja cijene gdje kupac ima potpunu kontrolu nad cijenom proizvoda ili usluge, uključujući opciju da ne plati ništa. Za razliku od drugih modela određivanja cijena kao što su aukcije ili model "odredi svoju cijenu" (*eng. Name-your-own-price*), u "Plati koliko želiš" modelu prodavatelj mora prihvatiti cijenu koju kupac odredi bez postavljanja praga ili mogućnosti odbijanja.

Ovaj model je specifičan po tome što omogućava kupcima da sudjeluju u procesu odre-

divanja cijene, što može dovesti do većeg osjećaja pravednosti i zadovoljstva. Također omogućava različite cijene na temelju individualnih procjena vrijednosti, čime privlači širi spektar kupaca, uključujući one koji bi inače bili isključeni zbog visoke cijene.

"Plati koliko želiš" djeluje unutar okvira "socijalnog tržišta", a ne "novčanog tržišta", gdje društvene norme kao što su uzajamnost i suradnja utječu na odluku kupca. Zanimljivo je da, iako imaju opciju da ne plate ništa, većina kupaca ipak plati pozitivnu cijenu, vođena društvenim normama i unutarnjim referentnim cijenama, a ne isključivo racionalnim ekonomskim ponašanjem.

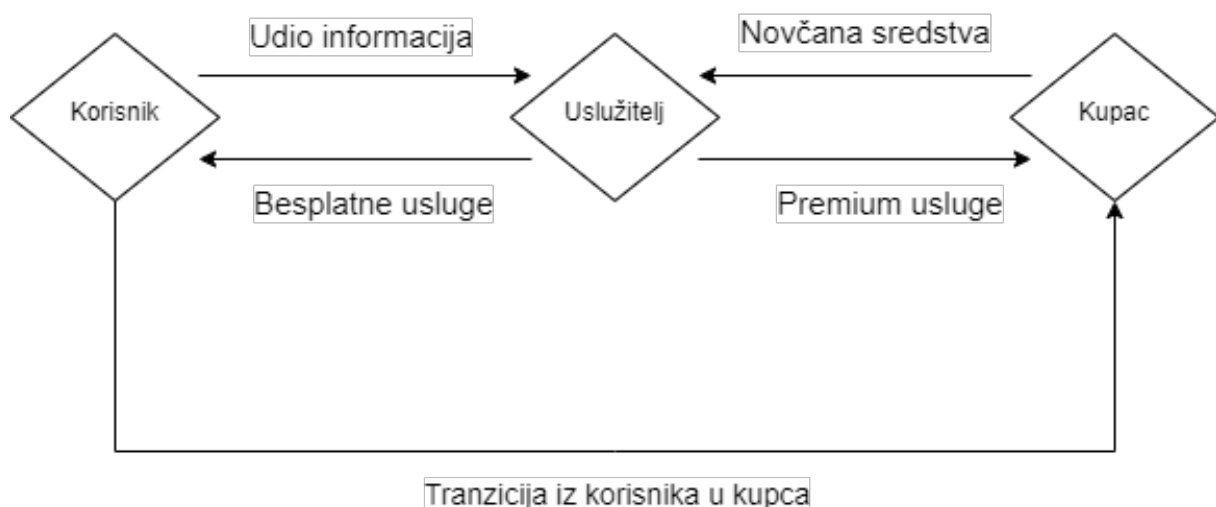
Istraživanja pokazuju da su potrošači često vođeni sjećanjima na prethodne cijene u nedostatku eksplicitne cijene, što dovodi do odluka o cijeni koje se temelje na njihovoj percepciji pravedne vrijednosti ili društvenim očekivanjima, umjesto na strogoj maksimalizaciji korisnosti.

4.3. Freemium

Freemium model [16] je model u kojem korisnik ne mora platiti proizvod kako bi ga koristio, ali daje mogućnosti nadoplate kako bi se koristilo bolje i više značajki proizvoda. Sam koncept zasniva se na pretpostavci kako će korisnik prilikom upoznavanja i istraživanja proizvoda odlučiti kupiti proizvod u cijelosti, odnosno postati *premium* korisnik.

Ovakav model nudi vrlo brzo penetriranje tržišta te je pogodan za mnoge današnje aplikacije poput Spotifyja, Discorda, Linked-ina i sl. Sve navedene platforme su lako dostupne svim korisnicima s poveznicom na internet, ali ako se korisnik odluči koristiti premium pogodnosti tada je obavezan platiti proizvod.

S nedavnim razvojem umjetne inteligencije može se pronaći velik broj SaaS modela umjetne inteligencije koji se financiraju putem freemium modela. Za primjer može se navesti ChatGPT koji se u ograničenom obujmu može koristiti za besplatno, no ako korisnik želi veću brzinu i integracije s ostalim modelima umjetne inteligencije potrebno je platiti za premium usluge.



Slika 1: Dijagram *Freemium* modela, vlastita izrada prema [16]

Na slici 1 vidljivo je kako korisnik može pristupati uslugama koje je uslužitelj (u ovom slučaju aplikacija) ponudio korisniku kao besplatne usluge. Te usluge korisnik najčešće plaća svojim podacima, odnosno podacima o ponašanju, preferencijama i slično. Ako se korisnik odluči za premium pogodnosti proizvoda tada se odvija sljedeća transakcija: Kupac daje novac u zamjenu za potpunu paletu usluga koje uslužitelj nudi.

Bitna stavka freemium modela je da uslužitelj unutar besplatnih usluga korisniku pokaže sve mogućnosti koje bi mogao imati ako odluči koristiti premium pogodnosti. Ovakav model poslovanja često se uparuje s cjenovnom strategijom pretplate kako bi se ostvarila maksimalna dobit.

5. Cjenovne strategije

Određivanje cjenovnih strategija ključna je točka u uspješnosti programskog proizvoda koji se prodaje. Kako bi proizvod ostvario puni potencijal bitno je detaljno istražiti tržište, njegove snage, slabosti, točke ulaska (*engl. Entry point*) i slično. Isto tako bitno je da se dobro poznaje sam programski proizvod koji se prodaje kako bi se odredile kompatibilnosti s određenim modelima cjenovnih strategija. Na primjer, želimo li prodavati uslugu poput strujanja (*engl. Streaming*) filmova ili prijenosa uživo tada bi bilo dobro razmotriti model pretplate. Ako se prodaje proizvod koji korisnicima omogućava usluge poput korištenja računalnog oblaka (*engl. Cloud computing*) ili skladišta podataka (*eng. Data Warehouse*), moguće je koristiti model plaćanja po korištenju kako bi obje strane bile zadovoljne; kupci plate onoliko koliko koriste, a s druge strane poslužitelj čuva i pametno raspoređuje resurse.

5.1. Pretplata

Model pretplate danas je najraširenija cjenovna strategija kod aplikacija koje nude usluge korisniku. Kod popularnih platformi za strujanje raznih videomaterijala poput Netflix, Internet Movie Database (IMDb) i sličnih, najlakše je implementirati ovu strategiju. Platforme za strujanje video materijala samo su jedan od primjera na koje je primjenjiva ova strategija. Pretplata se često kombinira s premium modelom kako bi se korisnicima omogućile razne pogodnosti na ograničeno vremensko trajanje, stoga taj model koriste i razni časopisi, online knjižnice, čak i platforme za postavljanje sadržaja poput Discorda.

U teoriji model pretplate donosi konstantan priljev novca poslužiteljima, ali bitno je prije toga dobro promisliti o pitanju: "Jesu li korisnici, odnosno "pretplatnici" voljni platiti za usluge koje poslužitelj nudi? Koliko su zapravo korisnici voljni odvojiti za usluge prije nego što odustanu od istih?". Premisa priljeva novca leži u tome da će sadržaj zadovoljiti korisnike i da će ih nastaviti zadovoljavati u budućnosti. U slučaju da oba kriterija nisu osigurana, potrebno je promijeniti cjenovnu strategiju. Preferabilne strategije koje su najčešća supstitucija pretplati su plaćanje po korištenju ili dinamično određivanje cijena ovisno o potražnji.

5.2. Plaćanje po korištenju

Plaćanje po korištenju (*engl. Pay-per-use*) cjenovna strategija oblik je poslovanja koji poslužitelju omogućava čuvanje resursa koje prodaje, dok korisniku omogućava da plati samo one resurse koje koristi. Najpoznatiji primjer ovog načina poslovanja može se vidjeti u poduzeću Amazon. Amazon Web Services jedna je od platformi koja svojim korisnicima nudi široku paletu usluga. Neke od usluga koje se naplaćuju putem modela plaćanja po korištenju su:

- Amazon EC2 [17] - Elastični računalni oblak (*eng. Elastic Cloud Computing*)
- Amazon S3 [18] - Jednostavna usluga skladištenja (*eng. Simple Storage Service*)

- AWS Lambda [19]
- Amazon RDS [20] - Usluga relacijske baze podataka (*eng. Relational database service*)
- Amazon API Gateway [21] - Pristupnik sučelja za programiranje aplikacija (*eng. Application programming interface gateway*)

Svaka od ovih usluga ima resurs prema kojemu se korisnicima naplaćuje tarifa korištenja. Elastični računalni oblak korisnicima naplaćuje količinu procesorske snage koju korisnik koristi po satu ili sekundi. Jednostavna usluga skladištenja naplaćuje se prema količini pohranjenih podataka, učestalosti pristupa skladištu i količini podataka koji se prenose. AWS Lambda naplaćuje vrijeme izvršavanja koda, odnosno broj zahtjeva za izvršavanje i trajanje izvršavanja koda. Usluga relacijske baze podataka naplaćuje se prema vrsti baza podataka, količine pohranjenih podataka i količini prometa podatcima. Pristupnik sučelja za programiranje aplikacija naplaćuje se na temelju broja zahtjeva prema pristupniku. Zajednički faktor ovim uslugama je da omogućuju korisniku fleksibilnost jer korisnik može prilagoditi troškove stvarnim potrebama i obujmu korištenja usluge. Ovaj model monetizacije najpopularniji je među poduzećima koja traže skalabilne i troškovno učinkovite usluge.

5.3. Dinamično određivanje cijena

Prilikom određivanja cijena postavlja se pitanje precjenjivanja ili podcjenjivanja proizvoda koji se prodaje. Određivanje cijene kompleksan je proces koji uzima u obzir veliku količinu faktora. Dinamično određivanje cijena taj problem rješava korištenjem prediktivnog modela koji uz pomoć velikih količina podataka kreira cijene za određeni trenutak vremena te se one konstantno mijenjaju.

Jedan takav primjer dinamičnog kreiranja cijena može se pronaći u osiguravajućem poduzeću Flock. [22] Flock svoje cijene osiguranja kreira uz pomoć podataka dobivenih kroz telematiku vozila (*engl. Telematics*). Isto tako sva vozila koja imaju ugrađene uređaje za praćenje podataka u vožnji su povezana putem poslužitelja (*engl. Server*) u kojem se ti podatci agregiraju i obrađuju u stvarnom vremenu. Cijene kreirane ovim putem vrlo su nestabilne te variraju za svakog vozača. Sigurniji vozači osiguranje plaćaju manje jer je rizik od nesreće znatno manji od vozača koji pokazuju znakove agresivne vožnje. Kako bi znali kakvo je ponašanje agresivnog vozača gledaju se podatci poput rapidnog povećanja ili smanjenja brzine, prosječne brzine i broja okretaja tijekom vožnje te mnogi drugi parametri prema kojima se određuje ponašanje vozača za volanom.

Drugi primjer dinamičnog generiranja cijena može se vidjeti kod generiranja cijena električne energije. Generiranje cijena u stvarnom vremenu (*eng. Real-Time Pricing (RTP)*) program je generiranja cijena električne energije u kojem sudionici programa svaki dan, ili sat, dobivaju cijene po kojima mogu kupiti električnu energiju. Najčešće je to na dnevnoj bazi, ali za velike klijente potrebno je kreirati cijene svakih sat vremena zbog velikih razlika u troškovima. [23]

5.4. Jednokratna naknada

Jednokratna naknada prilikom kupnje SaaS proizvoda odnosi se na jedinstveni iznos koji korisnik plaća unaprijed za određenu uslugu ili funkcionalnost unutar SaaS modela. Za razliku od mjesečnih ili godišnjih pretplata koje su tipične za SaaS proizvode, jednokratna naknada se plaća samo jednom i obično pokriva specifične troškove ili dodatne usluge. Prednosti ove cjenovne strategije je trenutna zarada za razliku od modela pretplate. Isto tako daje kupcima osjećaj vlasništva nad programskim proizvodom, a i osjećaj slobode jer klijent nije vezan ugovorom ili određenom vrstom pretplate na uslugu. Negativne posljedice ovog pristupa su nedostatak konstantnog priljeva novca tako da je teško planirati za razvoj programskog proizvoda jer u jednom trenutku taj instantni priljev novca neće biti dostatan ili će presušiti. Primjer ovakve cjenovne strategije može se naći u programskom proizvodu za uređivanje teksta naziva *Hemingway editor*. *Hemingway editor* nudi besplatnu *online* verziju s ograničenim setom značajki, ali isto tako nudi i aplikaciju za preuzimanje s cijenom od 19.99 dolara koja ima dodatne značajke koje nisu dostupne u *online* verziji. [24]

Jednokratna naknada ne mora isključivo biti za kupnju programskog proizvoda u cijelosti. Ovaj pristup može se koristiti za naplaćivanje implementacije koja uključuje troškove postavljanja, prilagodbe i integracije programskog proizvoda u postojeći sustav korisnika. Troškovi za obuku zaposlenika za korištenje programskog proizvoda ili troškovi početne podrške također se najčešće naplaćuju u obliku jednokratne naknade. [25]

5.5. Pretprodaja

Proces pretprodaje sastoji se od niza aktivnosti koje pomažu potencijalnim kupcima da shvate vrijednost proizvoda, steknu povjerenje i na posljetku donesu odluku o kupnji proizvoda. Usluga pretprodaje zapravo je posao koji se bazira na dijeljenju informacija, komunikacijom osjećaja i poboljšanjem stava potencijalnih kupaca prema proizvodu. Dvije značajke kojima se prožimaju svi procesi su besplatnost i interaktivnost. Potencijalnim kupcima daju se besplatni tehnički savjeti, besplatni interaktivni treninzi ili čak demonstracije same usluge ili aplikacije koja povezuje potencijalne kupce s njom [26]. Neke od ključnih aktivnosti u procesu pretprodaje su:

- Identifikacija i kvalifikacija potencijalnih kupaca
- Razumijevanje potreba kupca
- Prikaz proizvoda
- Tehnička podrška i savjetovanje
- Kreiranje i prezentacija vrijednosti

Kako bi pronašli potencijalne kupce prvo je bitno kvalitetno odrediti ciljanu publiku prema kojoj će se plasirati programski proizvod. Potrebno je razumjeti probleme koje programski proizvod rješava te potrebe potencijalnih kupaca kako bi se kvalitetno odabrala ciljana publika.

Unutar odabrane ciljane publike potrebno je klasificirati potencijalne klijente prema raznim kriterijima poput veličine poduzeća, budžet, hitnosti potrebe ili grani industrije. Klasificiranje se može vršiti i prema zainteresiranosti kupaca. Na primjer, u klasu A pripadali bi potencijalni kupci koji imaju jasnu namjeru kupnje proizvoda, unutar B klase nalazili bi se klijenti koji imaju generalnu namjeru kupnje programskog proizvoda, ali razmatraju dodatne opcije, a u klasi C nalazili bi se klijenti koji nemaju namjere za kupnju programskog proizvoda, ali su im potrebne konzultacije u području problema kojeg programski proizvod rješava. [26] Nakon jasno definiranih potencijalnih klijenata, bitno je razumjeti njihove potrebe. Prikupljanje podataka o klijentima uvelike olakšava definiranje njihovih poslovnih potreba, ali isto tako otvara i put za unaprjeđenje ili prilagođavanje programskog proizvoda kojeg im se plasira. Isto tako s klijentima se može ugovoriti sastanak u formi intervjua kako vi se razumjele specifični zahtjevi i izazovi s kojima se klijent susreće. Ako se klijentovim problemima pristupa s personaliziranim rješenjem koje odgovara njegovim problemima, velika je vjerojatnost da će taj klijent postati kupac tog proizvoda, a kako bi se lakše prikazalo klijentu da proizvod rješava njegove probleme i prilagođen je potencijalnom postojećem sustavu potrebno je provesti demonstraciju programskog proizvoda. U demonstraciji proizvoda dobro je istaknuti kako programski proizvod ponaša u specifičnim scenarijima koje kupac prolazi unutar svog poslovanja. Ako se klijent nakon uspješne demonstracije odluči na suradnju tada je potrebno napraviti tehničku procjenu kako bi se osiguralo da programski proizvod može biti integriran s postojećim sustavima, ako postoje, i biti prilagođen procesima klijenta. Nakon tehničke procjene može se napraviti dokaz koncepta (*eng. Proof of Concept (PoC)*) koji integrira osnovne značajke programskog proizvoda koji se prodaje u okruženju klijentovog poslovanja.

Aktivnost koja klijenta dodatno može potaknuti na kupnju je prezentacija dodatnih vrijednosti koje klijent ostvaruje korištenjem programskog proizvoda koji mu se prodaje. Na primjer, upotrebom studije slučaja klijentu se može prikazati kako će programski proizvod smanjiti troškove, povećati učinkovitost ili poboljšati korisničko iskustvo klijentovih korisnika/kupaca.

5.6. RazorBlade model

Razor-Blade model dobio je ime po poslovnom modelu pouzeća Gillette koje je osnovano još ranih 1900-tih godina. Model se zasniva na prodaji dva (ili više) komplementarnih proizvoda. Jedan proizvod se prodaje po nižoj cijeni kako bi privukao kupce, dok drugi proizvod koji je potreban za adekvatno korištenje prvog proizvoda. U primjeru Gillette poduzeća jedan proizvod je bila drška za brijanje dok je drugi, komplementarni proizvod bila sama britvica. Naime, sama drška prodavala se po vrlo povoljnim cijenama, ali su britvice bile relativno skupe.

Nedostatak ili prijetnja ovom modelu jest konkurencija, odnosno konkurentna poduzeća koja proizvode slične proizvode za istu namjenu. Kako bi poduzeća opstala na Razor-Blade monetizacijskom modelu koriste se svim trikovima kako bi zaštitila svoj proizvod, odnosno onemogućila konkurentnim poduzećima proizvodnju i prodaju istog. Neke od tehnika kojima se poduzeća služe su: zaštitni znakovi (*engl. Trademark*), patenti i ugovori o ekskluzivnoj prodaji. Primjeri ovakvog modela monetizacije česti su u fizičkim proizvodima poput pisača i njihovih ori-

ginalnih boja ili aparata za kavu i specifičnih kapsula koje odgovaraju samo navedenom aparatu za kavu. [27] Unutar IT industrije isto tako postoje primjeri ovog modela. Svako poduzeće koje je izdalo igraće konzole je puno veći profit ostvarilo na prodaji videoigara za svoju konzolu nego od prodaje samih konzola. Mobilna industrija također se zasniva na ovom modelu. Kupnja mobilnog uređaja nije dovoljna za njegovo potpuno korištenje jer mu je potrebna telekomunikacijska mreža koja se prodaje u paketima u poduzećima koja se bave isključivo telekomunikacijama. Bitan aspekt ovog monetizacijskog modela je mogućnost simbioze poduzeća koje proizvode komplementarne proizvode.

Velika prednost ovog modela je pristup kupcima koji nisu spremni potrošiti relativno velike količine sredstava kako bi probali proizvod te se samim time tržište vrlo brzo otvara. Isto tako, još jedna velika prednost očituje se u konstantnoj dobiti od proizvoda jer proizvod koji se prodaje po višoj cijeni je najčešće potrošan te ga kupac stalno mora kupovati kako bi nastavio koristiti proizvod.

Ovaj model monetizacije jako je sličan prethodno spomenutom *Freemium* modelu, ali bitna razlika je u tome kako se *Freemium* model zasniva na pretpostavci da će mali broj korisnika kupiti nadogradnju dok je kod RazorBlade modela pretpostavka da će većina kupaca kupiti komplementarni proizvod (nadogradnju) kako bi nastavila koristiti proizvod. [28]

6. Analiza primijenjenih oblika monetizacije na primjeru Internet Movie Database programskog proizvoda

Internet Movie Database (IMDb) je internetska baza podataka koja sadrži informacije o filmovima, televizijskim emisijama, glumcima, redateljima, scenaristima i drugim filmskim profesionalcima. Osnovao ju je Col Needham 1990. godine u obliku jednostavne liste filmova i glumaca, a danas je jedna od najpoznatijih i najkorištenijih baza podataka u filmskoj industriji.

IMDb omogućava korisnicima pristup širokom spektru informacija uključujući sinopsise filmova, detalje o produkciji, recenzije korisnika, ocjene filmova i slično. IMDb je u vlasništvu tvrtke Amazon od 1998. godine i koristi se ne samo od strane entuzijasta filmova, već i profesionalaca u industriji za prikupljanje i provjeru informacija o filmovima i osobama iz industrije. IMDb također nudi uslugu pod nazivom IMDbPro, koja je namijenjena profesionalcima i pruža dodatne informacije i alate za poslovnu upotrebu. IMDb je dostupan putem web stranice, ali isto tako i u obliku mobilne aplikacije te se smatra jednim od najpouzdanijih izvora informacija o filmskoj i televizijskoj industriji na globalnoj razini.

6.1. Oglasi

IMDb generira značajan dio svojih prihoda kroz prikazivanje oglasa na različitim dijelovima svoje platforme. Oglasi se pojavljuju na web stranici, u mobilnim aplikacijama, kao i unutar video sadržaja, poput najave filma i promotivnih materijala. Korisnici mogu naići na razne vrste reklama, uključujući bannere, pop-up prozore, video oglase prije sadržaja, te sponzorirane postove unutar rezultata pretrage i na profilima filmova. S obzirom na veliku posjećenost IMDb-a, oglašivači su spremni platiti premijske cijene za prikaz svojih oglasa korisnicima koji su zainteresirani za filmove i televiziju. Ovaj oblik oglašavanja cilja određene demografske skupine, što povećava efikasnost kampanja i prihod za IMDb.

6.2. IMDbPro Pretplate

IMDbPro je premium usluga koju IMDb nudi profesionalcima iz filmske i televizijske industrije. Pretplatnici na IMDbPro dobivaju pristup širokom rasponu dodatnih informacija koje nisu dostupne običnim korisnicima, uključujući detaljne informacije o filmovima u produkciji, detaljne profile industrijskih profesionalaca, te alate za praćenje trendova i analizu podataka. IMDbPro također omogućava korisnicima upravljanje svojim javnim profilima na IMDb-u, što je posebno korisno za glumce, redatelje i agente. Pretplate na IMDbPro naplaćuju se mjesečno ili godišnje, čineći ovu uslugu stabilnim izvorom prihoda za IMDb, dok istovremeno pruža dodanu vrijednost industriji.

6.3. Partnerstva i Licenciranje

IMDb ostvaruje prihod kroz partnerstva i licenciranje svog sadržaja drugim tvrtkama i platformama. Jedan od ključnih izvora prihoda dolazi od licenciranja IMDb-ove baze podataka putem API usluga, koje omogućuju drugim web stranicama, aplikacijama i medijskim platformama pristup informacijama o filmovima, serijama, glumcima i ekipama. Na primjer, razne platforme za streaming sadržaja, kao što su Netflix ili Hulu, koriste IMDb-ove podatke za prikazivanje informacija o naslovima dostupnim u njihovim bibliotekama. Osim toga, partnerstva s medijskim kompanijama omogućavaju IMDb-u da poveća svoju vidljivost i doda dodatne funkcionalnosti svojim uslugama, što također doprinosi prihodima.

6.4. Prodaja putem IMDb Trgovine

IMDb također ostvaruje prihod kroz svoj *online* dućan, IMDb Shop, gdje korisnici mogu kupovati razne proizvode povezane s filmovima i televizijom. U ovoj trgovini nalaze se artikli poput filmskih suvenira, majica, knjiga, DVD-ova i drugih srodnih proizvoda. Prodaja ovih proizvoda omogućava IMDb-u dodatne prihode, dok istovremeno nudi fanovima priliku da kupe predmete povezane s njihovim omiljenim filmovima i serijama. IMDb Shop također može sadržavati ekskluzivne proizvode, što dodatno potiče korisnike na kupovinu putem ove platforme.

6.5. Amazon Podružnički Program

Kao dio Amazonove obitelji, IMDb koristi prednosti Amazonovog podružničkog (*eng. affiliate*) programa za generiranje prihoda. Kada korisnici na IMDb-u kliknu na poveznice koje ih vode na Amazonovu stranicu za kupovinu, IMDb zarađuje proviziju od svake prodaje koja se ostvari putem navedenih poveznica. Ove poveznice često se nalaze na stranicama filmova, serija ili glumaca, gdje se korisnicima nude mogućnosti za kupovinu DVD-ova, Blu-ray diskova, knjiga, ili digitalnog sadržaja povezanog s tim naslovima. Ovaj oblik podružničkog marketinga omogućuje IMDb-u da dodatno unovči svoju veliku bazu korisnika i pruži im relevantne ponude za kupovinu, čime se osigurava dodatni prihod.

7. Praktični dio

U ovom praktičnom dijelu diplomskog rada prikazuje se izrada i implementacija mobilne aplikacije s pomoću .NET MAUI i Blazor tehnologija. .NET MAUI (Multi-platform App UI) je Microsoftova razvojna platforma koja omogućuje stvaranje aplikacija za više operacijskih sustava (Android, iOS, Windows i macOS) iz jedinstvene bazne programske strukture i koda. Blazor, s druge strane, nudi jednostavniji pristup kreiranju korisničkog sučelja korištenjem C# i .NET okruženja, zadržavajući sve prednosti web tehnologija. Integriranjem ove dvije tehnologije dobiva se moderna i izdržljiva arhitektura koja uvelike pojednostavljuje razvoj, održavanje i nadogradnju aplikacije.

U nastavku će biti detaljno objašnjena funkcionalna struktura aplikacije, uključujući ključne komponente sustava, način komunikacije s pozadinskim servisima te oblik i prikaz podataka korisniku. Posebna pažnja posvećena je prilagodbi sučelja različitim uređajima i operacijskim sustavima, što je jedna od glavnih prednosti korištenja .NET MAUI-a.

Cilj ovog praktičnog dijela je demonstrirati cjelokupni razvojni tijek – od inicijalnog postavljanja projekta i infrastrukturne pripreme, preko dizajna baze podataka i izrade korisničkog sučelja. Tako se jasno prikazuje kako kombinacija .NET MAUI-a i Blazora donosi efikasno, fleksibilno i održivo rješenje za razvoj modernih mobilnih aplikacija, pritom olakšavajući rad i skraćujući ukupno vrijeme razvoja.

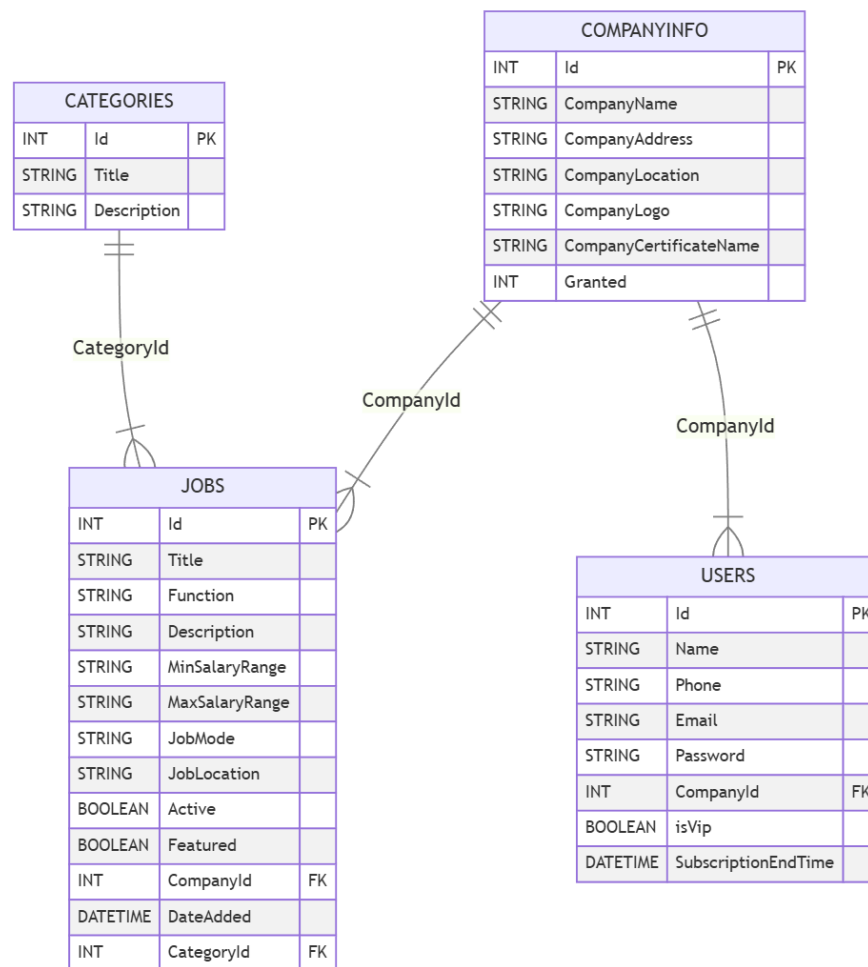
7.1. Baza podataka

U svrhu pohrane i upravljanja podacima u mobilnoj aplikaciji, odlučeno je koristiti SQLite bazu podataka zbog njezine jednostavnosti, male memorijske potrošnje i dobre integracije s .NET MAUI tehnologijom. SQLite je relacijski sustav za upravljanje bazama podataka (*eng. Relational DataBase Management System*) koji pohranjuje cijelu bazu unutar jedne datoteke. To ga čini idealnim rješenjem za mobilne aplikacije koje ne zahtijevaju kompleksne mrežne. Za potrebe kreiranja, pregleda i održavanja baze podataka korišten je alat SQLite Studio, koji nudi intuitivno sučelje za definiranje tablica, polja i relacija.

Unutar ove aplikacije, osnovna zadaća baze podataka je omogućiti brze i pouzdane operacije kreiranja, čitanja, ažuriranja i brisanja sadržaja (*eng. CRUD - Create, Read, Update, Delete*). Primjerice, prilikom kreiranja novog zapisa, podaci o korisniku ili nekom drugom entitetu upisuju se u odgovarajuću tablicu, dok čitanja operacija dohvaća sve relevantne podatke i prikazuje ih korisniku unutar sučelja aplikacije. Ažuriranje se vrši kada je potrebno izmijeniti postojeći zapis, dok se operacija brisanja koristi za brisanje podataka koji više nisu potrebni. Zahvaljujući jednostavnoj integraciji između .NET MAUI aplikacije i SQLite baze, ove su funkcionalnosti implementirane uz minimalan napor te se izvode izravno na mobilnom uređaju, bez potrebe za stalnom internetskom vezom.

U nastavku se detaljnije opisuje struktura tablica, definicija polja i relacije kreirane u bazi putem SQLite Studija, s primjerima koda koji prikazuju kako aplikacija komunicira s bazom. Na

taj način prikazat će se cjelovito rješenje za učinkovito upravljanje podacima u modernoj mobilnoj aplikaciji, pri čemu se posebno ističu prednosti jednostavne administracije i fleksibilnosti koje pruža SQLite.



Slika 2: Entitetsko-relacijski model, vlastita izrada.

U bazi podataka definirano je četiri tablice: *Categories*, *CompanyInfo*, *Jobs* i *Users*, koje zajedno čine osnovu za upravljanje informacijama o tvrtkama, poslovima te korisnicima u aplikaciji. Tablica *Categories* pohranjuje popis različitih kategorija (npr. "IT", "Marketing") te sadrži polja *Id*, *Title* i *Description*, omogućujući klasificiranje poslova radi lakšeg pretraživanja i organizacije. Tablica *CompanyInfo* služi za spremanje detalja o tvrtkama (npr. naziv, adresa, lokacija, logo i certifikati), tvrtka se identificira primarnim ključem *Id*, a svi povezani korisnici i poslovi referenciraju se preko atributa *CompanyId*.

Tablica *Jobs* omogućuje upravljanje oglasima za radna mjesta te sadrži podatke poput naziva posla, funkcije, opisa, raspona plaće, načina i lokacije rada, kao i oznake je li oglas aktivan ili istaknut. Dodatno, polja *CompanyId* i *CategoryId* služe kao strani ključevi, čime se svaki posao povezuje s konkretnom tvrtkom i kategorizira unutar definiranih kategorija.

Tablica *Users* obuhvaća sve korisnike aplikacije te, uz uobičajene stupce (npr. Name, Email, Phone i Password), sadrži i polja: *isVip* (logička vrijednost) i *SubscriptionEndTime* (da-

tum i vrijeme isteka pretplate). Tablica *Users* pomoću ta dva atributa upravlja razinama dostupnosti informacija u aplikaciji i pogodnostima korisnika s obzirom na njihovu razinu prava.

7.1.1. Kod baze

Implementacija baze realizirana je kroz klasu *DbService* koja predstavlja središnje mjesto za sve aktivnosti povezane s bazom podataka unutar aplikacije. Ova klasa temelji se na asinkronom radu sa SQLite-om kako bi se postiglo bolje rukovanje operacijama unosa, ažuriranja i brisanja podataka. Unutar svog konstruktora, klasa inicijalno stvara vezu (*SQLiteAsyncConnection*) prema lokalnoj bazi podataka, pri čemu se lokacija određuje kombiniranjem statički definiranog naziva baze i putanje dobivene iz *FileSystem.AppDataDirectory* datoteke. Jedna od najvažnijih metoda jest *InitializeAndSeedAllAsync()*, koja služi za početno postavljanje baze i testnih podataka. Najprije se, ako postoje, brišu sve tablice, zatim se one nanovo stvaraju te se unose početni zapisi radi testiranja, uključujući kategorije, tvrtke, korisnike, ali i razne poslove s pripadajućim opisima, rasponima plaća i kategorijama. U klasi se isto tako nalaze CRUD metode za svaku od relevantnih tablica, omogućujući razvijatelju da brzo i lako upravlja zapisima u bazi. Također, postoje i metode za dohvat svih zapisa unutar pojedine tablice. Posebno je korisna metoda *AuthenticateUser*, koja na temelju korisničke e-pošte i lozinke provjerava postoji li odgovarajući zapis u tablici *User*, čime je omogućen osnovni mehanizam autentikacije unutar aplikacije. Na taj način, *DbService* objedinjuje logiku početne inicijalizacije, ponovnog postavljanja i populiranja baze podacima te praktične metode za rukovanje podatkovnim modelima, stavljajući naglasak na asinkrone operacije koje poboljšavaju odzivnost aplikacije i cjelokupno korisničko iskustvo.

Isječak koda 1: Isječak koda baze, vlastita izrada

```
1 private const string dbName = "BazaDb.db";
2 private readonly SQLiteAsyncConnection _connection;
3
4 #Konstruktor klase
5 public DbService()
6 {
7     var dbPath = Path.Combine(FileSystem.AppDataDirectory, dbName);
8     _connection = new SQLiteAsyncConnection(dbPath);
9 }
10
11 #AuthenticateUser funkcija za provjeravanje korisnika prilikom prijave u aplikaciju
12 public async Task<User> AuthenticateUser(string email, string password) =>
13     await _connection.Table<User>()
14         .Where(u => u.Email == email && u.Password == password)
15         .FirstOrDefaultAsync();
16
17 #Primjer CRUD funkcija unutar baze za klasu User
18 public async Task<int> InsertUser(User user) =>
19     await _connection.InsertAsync(user);
20
21 public async Task<int> UpdateUser(User user) =>
22     await _connection.UpdateAsync(user);
23
```

```

24 public async Task<int> DeleteUser(User user) =>
25     await _connection.DeleteAsync(user);
26
27 public async Task<User> GetUserById(int id) =>
28     await _connection.Table<User>().Where(u => u.Id == id).FirstOrDefaultAsync();

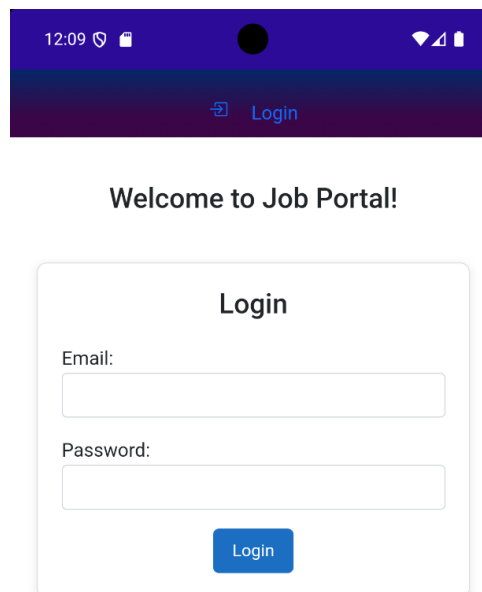
```

7.2. Stranice aplikacije

U sklopu praktičnog dijela aplikacije razvijene su četiri ključne stranice koje zajedno tvore zaokruženu cjelinu funkcionalnosti: *Login*, *Jobs*, *CreateJobPosting* i *Profile*. Svaka stranica ima specifičnu svrhu i međusobno se nadopunjuju kako bi korisnicima omogućile cjelovito iskustvo prijave, pregleda, kreiranja i osobnog upravljanja vlastitim podacima. U nastavku se detaljnije objašnjava uloga svake stranice, njezin izgled i način na koji komunicira s bazom i poslužiteljem. Uz opis same aplikacije postaviti će se i slike zaslona svake stranice te isječki kodova bitnih dijelova stranica.

7.2.1. Prijava

Stranica *Login.cs* predstavlja početnu točku aplikacije, što je unutar koda označeno dekoratorom `@page "/"`, i omogućuje korisniku jednostavan mehanizam za prijavu pomoću njegove e-mail adrese i lozinke. Unutar klase pozivaju se ključni servisi za funkcioniranje aplikacije: *DbService*, koristi se za pristup i manipulaciju bazom podataka, a *NavigationManager* upravlja navigacijom unutar Blazor sučelja, naposljetku, *UserSessionService* pohranjuje informacije o trenutno prijavljenom korisniku, čime omogućuje globalno rukovanje korisničkom sesijom. Glavni dio stranice sastoji se od zaglavlja s naslovom “Welcome to Job Portal!” te ele-



Slika 3: Slika zaslona stranice za prijavu u aplikaciju, vlastita izrada

menta koji sadrži naslov “Login” i obrazac za unos korisničkih podataka. Sam obrazac izrađen je s pomoću Blazor komponente `<EditForm>` i ima svoj Model koji se u ovom slučaju naziva `loginModel`, te atribut `OnValidSubmit` koji poziva metodu `HandleLogin` kada se forma ispravno popuni i pošalje. U `<EditForm>` su uključene ugrađene komponente za validaciju, točnije `DataAnnotationsValidator` i `ValidationSummary`, s pomoću kojih se ispisuju eventualne poruke o pogrešnim ili nepotpunim unosima. Unutar forme, polja za `Email` i `Password` vežu se na model `loginModel` preko direktive `@bind-Value`.

Ako se ispostavi da se unesenim podacima podudara zapis u bazi, što se provjerava metodom `AuthenticateUser` unutar `DbService` klase, korisnik se sprema u svojstvo `UserSession.LoggedUser`, a zatim se uz pomoć `Navigation.NavigateTo("/jobs")` preusmjerava na rutu `/jobs` koja simbolizira prelazak na novu stranicu koja sadrži popis poslova. U suprotnom, putem svojstva `ErrorMessage`, prikazuje se poruka o nevažecim vjerodajnicama. Konačno, kako bi se olakšalo rukovanje potrebnim vrijednostima za prijavu, unutar `@code` bloka definirana je klasa `LoginModel` s dvama glavnim atributima, `Email` i `Password`, koji mapiraju odgovarajuće vrijednosti u obrascu.

Isječak koda 2: Kod stranice za prijavu, vlastita izrada

```

1 @page "/"
2 @inject DbService DbService
3 @inject NavigationManager Navigation
4 @inject JobPortalMAUI.Services.UserSessionService UserSession
5
6 <div class="login-title">
7     <h3>Welcome to Job Portal!</h3>
8 </div>
9 <div class="login-page-container">
10     <h3>Login</h3>
11
12     @if (!string.IsNullOrEmpty(ErrorMessage))
13     {
14         <div class="alert alert-danger">@ErrorMessage</div>
15     }
16
17     <EditForm Model="loginModel" OnValidSubmit="HandleLogin">
18         <DataAnnotationsValidator />
19         <ValidationSummary />
20         <div class="form-group">
21             <label>Email:</label>
22             <InputText @bind-Value="loginModel.Email" class="form-control" />
23         </div>
24         <div class="form-group">
25             <label>Password:</label>
26             <InputText @bind-Value="loginModel.Password" type="password" class="form-control" />
27         </div>
28         <button type="submit" class="btn btn-primary">Login</button>
29     </EditForm>
30 </div>
31

```

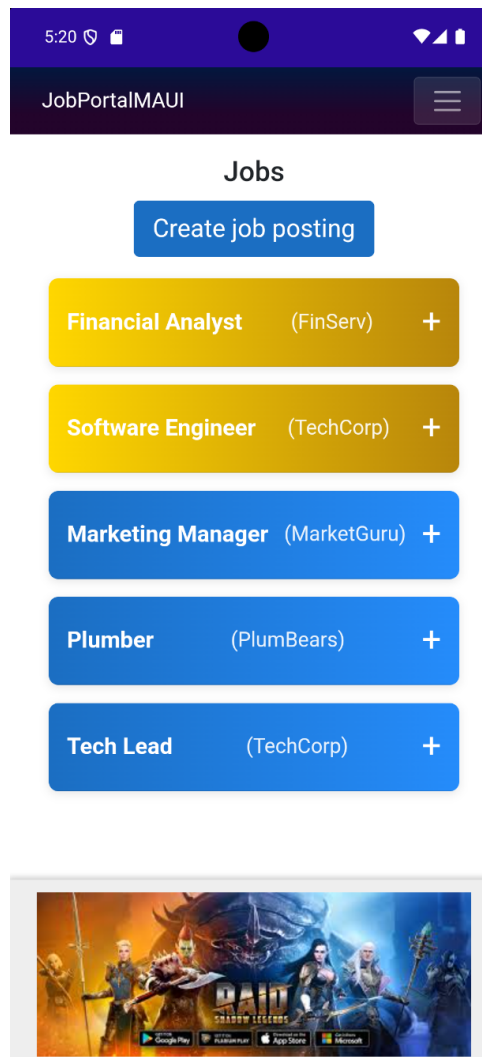
```

32 @code {
33     private LoginModel loginModel = new LoginModel();
34     private string ErrorMessage;
35
36     private async Task HandleLogin()
37     {
38         ErrorMessage = string.Empty;
39         var user = await DbService.AuthenticateUser(loginModel.Email, loginModel.
40             Password);
41         if (user != null)
42         {
43             UserSession.LoggedUser = user;
44             Navigation.NavigateTo("/jobs");
45         }
46         else
47         {
48             ErrorMessage = "Invalid email or password.";
49         }
50     }
51
52     public class LoginModel
53     {
54         public string Email { get; set; }
55         public string Password { get; set; }
56     }

```

7.2.2. Poslovi

Stranica *Jobs.razor* (označena dekoratorom *@page "/jobs"*) prikazuje popis oglasa za radna mjesta, uz niz informacija vezanih uz sami oglas. Na početku se pozivaju (*eng. inject*) servisi *DbService*, *UserSessionService* i *NavigationManager*, čime se omogućuje dohvat zapisa iz baze podataka, upravljanje podacima o trenutno prijavljenom korisniku te preusmjerenje na druge stranice aplikacije. Na samom vrhu stranice, ispod naslova "Jobs", nalazi se gumb za kreiranje novog oglasa "Create job posting" koji je vidljiv samo ako je korisnik prijavljen i ima pripadajući *CompanyId*, odnosno već radi za određenu tvrtku.



Slika 4: Slika zaslona stranice oglasa poslova, vlastita izrada

Glavna logika prikaza oglasā smještena je unutar elementa s klasom *job-list-container*, gdje se najprije prikazuje poruka “*Loading jobs...*” dok se podaci učitavaju iz baze, odnosno “*No jobs found.*” ako nema nijednog zapisa. Nakon što se dohvate poslovi, ime i kontakt podaci tvrtke, iterira se kroz kolekciju *jobs* te se za svaki posao iscrtava odgovarajuća kartica koja sadrži jedan oglas za posao i ikonu za proširivanje ili sažimanje detalja o poslu. U proširenom dijelu prikazuju se informacije kao što su nazivi funkcija, opis, raspon plaće i datum dodavanja, a stupci *Phone* i *Email* tvrtke dohvaćaju se putem funkcija *GetContactPhone* i *GetContactEmail*.

Zanimljiv detalj jest promjenjiva razina prikaza informacija, koja se određuje zastavicom (eng. *flag*) *isDetailed*. Ako korisnik ima aktivnu pretplatu, prikazuju se dodatne informacije o poslu, dok su inače dostupni samo osnovni podatci. Sam dohvat poslova i pripadajućih korisnika realizira se u funkciji *OnInitializedAsync*, gdje se zapisi sortiraju tako da poslovi sa zastavicom *Featured* dolaze prvi i zlatne su boje. Na kraju, klikom na gumb “*Create job posting*” poziva se metoda *CreateJobPosting* koja vodi korisnika na stranicu za kreiranje oglasa posla, podržavajući tako cjelokupni ciklus rada s poslovima unutar ove stranice.

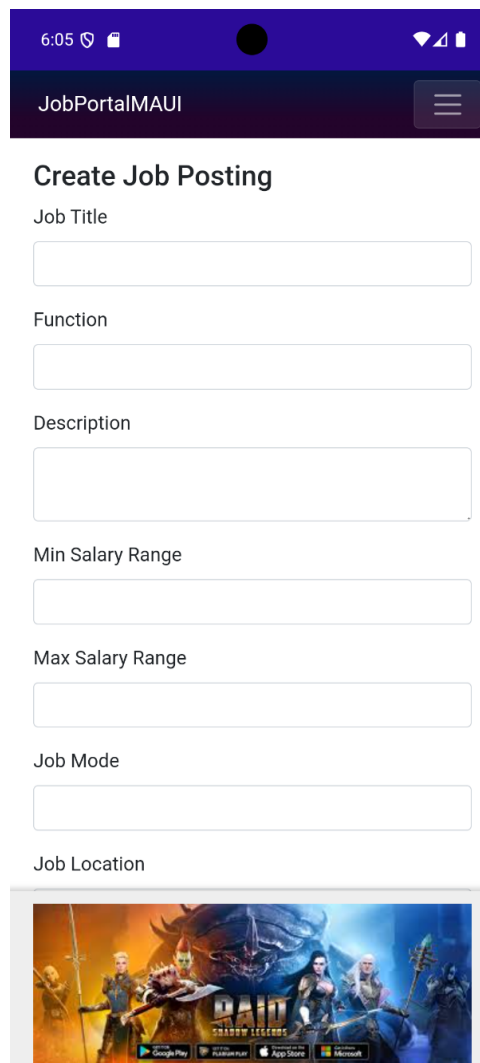
Isječak koda 3: Kod stranice za prikaz poslova, učitavanje podataka, vlastita izrada

```
1 <div class="job-list-container">
2     @if (jobs == null || companyDict == null || contactDict == null)
3     {
4         <p>Loading jobs...</p>
5     }
6     else if (!jobs.Any()) { <p>No jobs found.</p> }
7     else
8     {
9         @foreach (var job in jobs)
10        {
11            <div class="job-item card">
12                <div class="job-title card-header @(job.Featured ? "featured" : "")"
13                    @onclick="@(() => ToggleJob(job))">
14                    <strong>@job.Title</strong>
15                    <span class="company-name">(@GetCompanyName(job.CompanyId))</span>
16                    <span class="toggle-icon">@((job.IsExpanded) ? "-" : "+")</span>
17                </div>
18                <div class="job-details card-body @(job.IsExpanded ? "expanded" : "
19                    collapsed")">
20                    @if (isDetailed)
21                    {
22                        <p><strong>Function:</strong> @job.Function</p>
23                        <p><strong>Description:</strong> @job.Description</p>
24                        <p><strong>Min Salary:</strong> @job.MinSalaryRange</p>
25                        <p><strong>Max Salary:</strong> @job.MaxSalaryRange</p>
26                        <p><strong>Job Mode:</strong> @job.JobMode</p>
27                        <p><strong>Job Location:</strong> @job.JobLocation</p>
28                        <p><strong>Active:</strong> @job.Active</p>
29                        <p><strong>Date Added:</strong> @job.DateAdded.ToString("g")
30                        </p>
31                    }
32                    else
33                    {
34                        <p><strong>Description:</strong> @job.Description</p>
35                    }
36                <p>
37                    <strong>Contact Phone:</strong>
38                    <i class="fas fa-phone"></i> @GetContactPhone(job.CompanyId)
39                </p>
40                <p>
41                    <strong>Contact Email:</strong>
42                    <i class="fas fa-envelope"></i> @GetContactEmail(job.
43                    CompanyId)
44                </p>
45            </div>
46        }
47    }
48 </div>
```

7.2.3. Kreiranje oglasa za posao

Stranica za kreiranje poslova korisnicima koji su povezani s tvrtkom omogućuje kreiranje novih oglasa za posao. Na početku se pozivaju servisi *DbService* za spajanje s bazom podataka, *UserSessionService* koja provjerava je li korisnik prijavljen te *NavigationManager* za kretanje korisnika po aplikaciji. Tijekom inicijalizacije provjerava se ima li trenutno ulogirani korisnik pridruženu tvrtku, a zatim se dohvaća popis kategorija kako bi se prikazale u padajućem izborniku.

U slučaju da korisnik nema valjanu tvrtku ne može pristupiti ovoj stranici što je implementirano tako da gumb za kreiranje poslova uopće nije vidljiv. Usporedno se inicijalizira objekt *jobModel*, u koji se spremaju informacije koje korisnik može unijeti na stranici. Na samoj stranici prikazuje se obrazac s poljima za naslov, funkciju, opis, raspon plaće i druge relevantne atribute. Za polje kategorije iskorišten je element padajućeg izbornika koji prikazuje stavke iz dohvaćene liste kategorija. Ako korisnik označi oglas kao “*Featured*”, unutar forme se ispisuje poruka o naknadi koju korisnik mora platiti ako želi objaviti oglas.



6:05

JobPortalMAUI

Create Job Posting

Job Title

Function


Description

Min Salary Range

Max Salary Range

Job Mode

Job Location

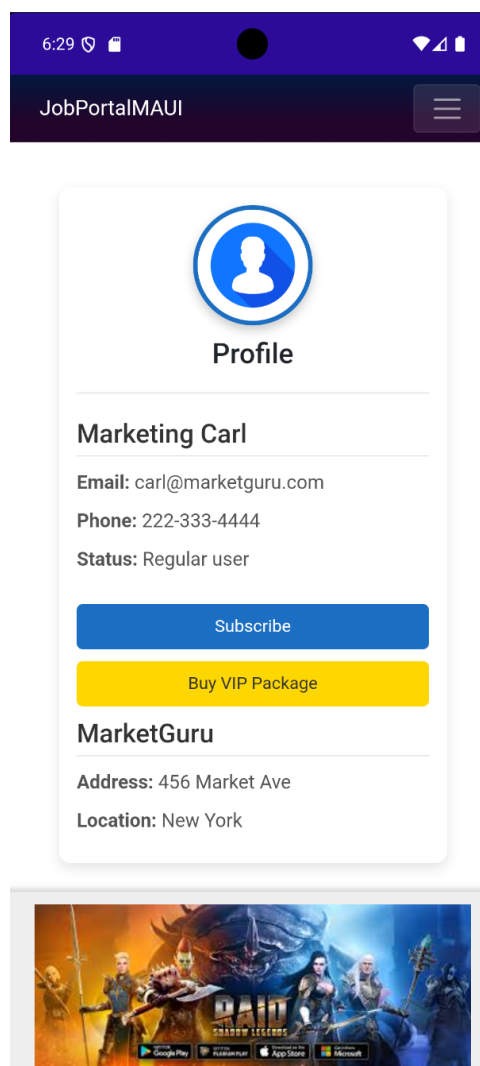


Slika 5: Slika zaslona stranice za kreiranje oglasa, vlastita izrada

Validiran i uspješno popunjen obrazac ubacuje se u bazu metodom *DbService.InsertJob*, uz prethodnu provjeru želi li korisnik zaista imati istaknut (*eng. featured*) oglas. Kada korisnik klikne na gumb "Submit", prikaže se prozor u kojem se nalazi poruka koja govori korisniku da je uspješno objavio oglas. Nakon toga, korisnik se preusmjerava natrag na stranicu s popisom poslova, a u slučaju da želi odustati od kreiranja oglasa, dostupan je gumb "Cancel" koji također vodi na stranicu poslova.

7.2.4. Profil korisnika

Stranica profila služi za prikaz i ažuriranje ključnih korisničkih podataka unutar aplikacije. U slučaju da korisnik posjeduje valjani korisnički račun, prikazuju se njegovo ime, adresa e-pošte, telefonski broj te trenutni status. Statusi koji trenutno postoje su: "Regular user", "Subscribed user", "VIP user" i "VIP Subscriber". Status korisnika dohvaća se s pomoću metode *GetUserStatusText*. K tome, ako je korisnik povezan s nekom tvrtkom, dohvaćaju se podaci te tvrtke iz baze i prikazuju njezino ime, adresa i lokacija. Ako takva poveznica ne postoji, ispisuje se poruka da korisnik ne pripada nijednoj tvrtki.



Slika 6: Slika zaslona stranice profila, vlastita izrada

Na stranici profila korisnik ima mogućnost pretplate (*eng. Subscribe*) na aplikaciju kako bi ostvario dodatne pogodnosti. Kada se pretplati, korisniku se u bazi postavlja vrijeme isteka pretplate uz istodobno prikazivanje skočnog prozora na kojem je prikazana obavijest da je korisnik uspješno pretplaćen na korištenje aplikacije. Ako korisnik nema *VIP* status, može se odlučiti za nadogradnju, koja postavlja polje *IsVIP* na "true" i obavještava ga o uspješnoj kupnji *VIP* statusa.

Kod klika na gumbе za pretplatu ili kupnju *VIP* statusa, *UserSession* se osvježava novim zapisima iz baze i prikazuje poruka o uspješnoj operaciji. Na taj način stranica profila osigurava jednostavno i intuitivno upravljanje korisničkim informacijama te pruža opcije za nadogradnju statusa za dodatne pogodnosti.

Isječak koda 4: Kod stranice profila, vlastita izrada

```
1 <div class="profile-container">
2 <div class="profile-header">
3 
5 <h3>Profile</h3>
6
7 @if (UserSession.LoggedUser == null)
8 {
9 <p>You are not logged in. Please <NavLink href="/">login</NavLink>.</p>
10 }
11 else
12 {
13 <div class="user-details">
14 <h4>@UserSession.LoggedUser.Name</h4>
15 <p><strong>Email:</strong> @UserSession.LoggedUser.Email</p>
16 <p><strong>Phone:</strong> @UserSession.LoggedUser.Phone</p>
17 <p><strong>Status:</strong> @GetUserStatusText ()</p>
18
19 <div class="button-group">
20 @if (UserSession.LoggedUser.SubscriptionEndTime == null ||
21 UserSession.LoggedUser.SubscriptionEndTime <= DateTime.Now)
22 {
23 <button class="btn btn-primary" @onclick="Subscribe">
24 Subscribe</button>
25 }
26 @if (!UserSession.LoggedUser.IsVIP)
27 {
28 <button class="btn vip-btn" @onclick="BuyVIP">Buy VIP
29 Package</button>
30 }
31 </div>
32 </div>
33
34 @if (UserSession.LoggedUser.CompanyId.HasValue && UserSession.LoggedUser.
35 CompanyId.Value != 0)
36 {
37 @if (companyInfo == null)
```

```

35     {
36         <p>Loading company information...</p>
37     }
38     else
39     {
40         <div class="company-details p">
41             <h4>@companyInfo.CompanyName</h4>
42             <p><strong>Address:</strong> @companyInfo.CompanyAddress</p>
43             <p><strong>Location:</strong> @companyInfo.CompanyLocation</p>
44         </div>
45     }
46 }
47 else
48 {
49     <p class="no-company">User does not work in any company.</p>
50 }
51 }
52 </div>

```

7.3. Primijenjene tehnike monetizacije aplikacije

U sklopu aplikacije ponuđena su četiri različita modela monetizacije, čime se osigurava financijska održivost sustava i potiču dodatne pogodnosti za korisnike. Prvi model je pretplata, pri čemu korisnik uplaćuje određeni iznos kako bi na određeno vrijeme dobio pristup detaljnijim informacijama o poslovima i imao bolji uvid u posao kojeg traži. Unutar aplikacije realiziran je s pomoću vremenskog pomaka nakon korisnikovog odabira pretplate koja se nalazi na stranici korisničkog profila ispod samih informacija o korisniku.

Drugi je VIP paket, koji podrazumijeva trajni pristup ekskluzivnim funkcionalnostima, poput funkcioniranja aplikacije bez nepotrebnih reklama, bez potrebe da se korisnik pretplaćuje iznova. U ovom slučaju VIP paket prikazan je u obliku uklanjanja reklama iz aplikacije ako je korisnik VIP član aplikacije. Ova opcija prikazuje model jednokratne kupnje te nudi primamljivu priliku korisniku da ukloni nepoželjne reklame iz aplikacije za malu svotu novca. Ovakav način monetizacije često se može vidjeti uparen s reklamama kao i u slučaju ove aplikacije.

Treći model su istaknuti oglasi, gdje tvrtke plaćaju određenu naknadu kako bi se njihovi oglasi posebno naglasili i pri samom vrhu rezultata, privlačeći time više kandidata. Istaknuti oglasi pripadaju modelu plaćanja po korištenju jer je korisnik primoran za svaki oglas koji želi istaknuti platiti određenu svotu, ali nije obavezan to učiniti za svaki oglas koji postavi. Ovaj pristup daje korisnicima slobodu izbora da reguliraju svoje uplate kako oni to žele i plate samo kada im je to potrebno.

Četvrti model su prostori za reklamne sadržaje unutar aplikacije. Sam prostor na zaslonu može se prodati u reklamne svrhe tvrtkama direktno ili jednostavno izdavaču aplikacije dati u najam kako bi izdavač regulirao koji sadržaj će se reklamirati unutar aplikacije. U slučaju da izdavač ima pristup sadržaju tada se najčešće stavljaju oglasi usko vezani uz aplikaciju kako bi što više korisnika kliknulo na reklamni sadržaj.

Ova četiri sustava monetizacije zajedno nude fleksibilnost i za korisnike koji traže posao i za tvrtke koje žele istaknuti svoje natječaje, dok se istovremeno generiraju financijska sredstva potrebna za daljnje održavanje i razvoj platforme.

8. Zaključak

U ovome radu prikazani su različiti načini monetizacije programskih proizvoda, počevši od općih koncepata i strategija, sve do specifičnih modela implementiranih u praktičnom dijelu. Analiza je pokazala da izbor odgovarajuće kombinacije oblika naplate i dodatnih pogodnosti (poput pretplate, VIP statusa ili isticanja oglasa) uvelike ovisi o samoj prirodi proizvoda i potrebama krajnjih korisnika. U teorijskom dijelu rada objašnjene su ključne značajke modela kao što su freemium, plaćanje po korištenju i dinamično određivanje cijena, dok su u praktičnom dijelu jasno demonstrirane mogućnosti njihove integracije na konkretnoj aplikaciji rađenoj u .NET MAUI/Blazor okruženju.

Tako razvijeno rješenje ne samo da olakšava upravljanje poslovima i korisnicima, nego istodobno dokazuje da monetizacijske tehnike mogu biti uspješno uključene i u jednostavnu mobilnu aplikaciju. Time su potvrđene prednosti cjelovitog pristupa, u kojem teorijska znanja o tržištu i cjenovnim strategijama nadopunjuju praktičan rad s bazom podataka, korisničkim sučeljem i poslovnom logikom. U konačnici, radna hipoteza o važnosti pomne selekcije i prilagodbe načina monetizacije potvrđena je primjerima, sugerirajući da je ključ održivog financijskog modela ravnoteža između korisničkih očekivanja, tehničke izvedivosti te jasno definiranih poslovnih ciljeva.

Popis literature

- [1] A. Payne, P. Frow i A. Eggert, „The customer value proposition: evolution, development, and application in marketing,” *Journal of the Academy of Marketing Science*, sv. 45, str. 467–489, 2017.
- [2] *What innovative approaches are SaaS companies taking to monetize data while ensuring privacy and compliance? - Software-as-a-Service* — *flevy.com*, <https://flevy.com/topic/software-as-a-service/question/innovative-saas-monetization-strategies-ensuring-data-privacy-compliance>, [Accessed 17-08-2024].
- [3] *Managed DDos Protection - AWS Shield - AWS* — *aws.amazon.com*, <https://aws.amazon.com/shield/>, [Accessed 17-08-2024].
- [4] *Intelligent Threat Detection – Amazon GuardDuty – AWS* — *aws.amazon.com*, <https://aws.amazon.com/guardduty/>, [Accessed 17-08-2024].
- [5] E. L. Neely, „Come for the game, stay for the cash grab: The ethics of loot boxes, microtransactions, and freemium games,” *Games and Culture*, sv. 16, br. 2, str. 228–247, 2021.
- [6] N. Ballou, C. T. T. Gbadamosi i D. Zendle, „The hidden intricacy of loot box design: A granular description of random monetized reward features,” *PsyArXiv. doi*, sv. 10, 2020.
- [7] *Blazingly fast email for teams and individuals*.
- [8] J. Mehta, „How to create a sense of exclusivity on Your saas landing page,” *Abmatic AI*, 11. 2023.
- [9] A. Ashwini, „How to set up a viral waiting list for your SAAS product,” *DataDab Insights*, 2024.
- [10] *Steam Store* — *store.steampowered.com*, <https://store.steampowered.com/>, [Accessed 20-08-2024].
- [11] *Discord - Group Chat That's All Fun & Games* — *discord.com*, <https://discord.com/>, [Accessed 20-08-2024].
- [12] European Parliament i Council of the European Union, *Regulation (EU) 2016/679 of the European Parliament and of the Council*, of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ L 119, 4.5.2016, p. 1–88, 4. 5. 2016.

- [13] P. BUKATY, *The California Privacy Rights Act (CPRA) – An implementation and compliance guide*. IT Governance Publishing, 2021., ISBN: 9781787782860.
- [14] *Signal Messenger: Speak Freely* — signal.org, <https://signal.org/>, [Accessed 20-08-2024].
- [15] R. Roy, „An insight into pay-what-you-want pricing,” *Marketing Intelligence & Planning*, sv. 33, br. 5, str. 733–748, 2015.
- [16] N. Pujol, „Freemium: attributes of an emerging business model,” *Available at SSRN 1718663*, 2010.
- [17] *Amazon EC2 - Cloud Compute Capacity - AWS* — [aws.amazon.com](https://aws.amazon.com/ec2/), <https://aws.amazon.com/ec2/>, [Accessed 20-08-2024].
- [18] *Amazon S3 - Cloud Object Storage - AWS* — [aws.amazon.com](https://aws.amazon.com/s3/), <https://aws.amazon.com/s3/>, [Accessed 20-08-2024].
- [19] *Serverless Function, FaaS Serverless - AWS Lambda - AWS* — [aws.amazon.com](https://aws.amazon.com/lambda/), <https://aws.amazon.com/lambda/>, [Accessed 20-08-2024].
- [20] *Managed SQL Database - Amazon Relational Database Service (RDS) - AWS* — [aws.amazon.com](https://aws.amazon.com/rds/), <https://aws.amazon.com/rds/>, [Accessed 20-08-2024].
- [21] *API Management - Amazon API Gateway - AWS* — [aws.amazon.com](https://aws.amazon.com/api-gateway/), <https://aws.amazon.com/api-gateway/>, [Accessed 20-08-2024].
- [22] F. Poduzeće, *Connected Fleet Insurance*, 2015.
- [23] A. Faruqi, R. Hledik i J. Tsoukalis, „The power of dynamic pricing,” *The Electricity Journal*, sv. 22, br. 3, str. 42–56, 2009.
- [24] P. Šobak, *SAAS pricing models explained (with examples)*, 3. 2024.
- [25] P. Wolf, *Founder's Cheat Sheet to SAAS pricing models and strategies*, 5. 2024.
- [26] F.-I. Chen, „Analysis on Knowledge Framework and Development Trend of the Presale Services,” *2010 International Conference of Information Science and Management Engineering*, sv. 1, 2010., str. 295–298. DOI: 10.1109/ISME.2010.121.
- [27] W. Kenton, *Razor-Razorblade Model: Definition, how it works, and examples*, 12. 2022.
- [28] S. Sloth, *The Razor-Razorblade Business Model: Examples of the sales approach*, 4. 2021.

Popis slika

1.	Dijagram <i>Freemium</i> modela, vlastita izrada prema [16]	13
2.	Entitetsko-relacijski model, vlastita izrada.	23
3.	Slika zaslona stranice za prijavu u aplikaciju, vlastita izrada	25
4.	Slika zaslona stranice oglasa poslova, vlastita izrada	28
5.	Slika zaslona stranice za kreiranje oglasa, vlastita izrada	30
6.	Slika zaslona stranice profila, vlastita izrada	31

Popis isječaka koda

1.	Isjeak koda baze, vlastita izrada	24
2.	Kod stranice za prijavu, vlastita izrada	26
3.	Kod stranice za prikaz poslova, učitavanje podataka, vlastita izrada	29
4.	Kod stranice profila, vlastita izrada	32