

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

ZAKLJUČNA STROKOVNA NALOGA VISOKE POSLOVNE ŠOLE

RAZVOJ PROGRAMSKE REŠITVE ZA NAROČANJE NA STORITVE

Ljubljana, september 2022

GAŠPER PEČNIK

IZJAVA O AVTORSTVU

Podpisani Gašper Pečnik, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Razvoj programske rešitve za naročanje na storitve, pripravljenega v sodelovanju s svetovalcem red. prof. dr. Tomažem Turkom

IZJAVLJAM

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne _____

Podpis študenta: _____

KAZALO

UVOD	1
1 APLIKACIJA	2
1.1 Definicija aplikacije	2
1.2 Življenjski cikel razvoja aplikacije	3
1.3 Predstavitev različnih modelov razvoja aplikacij	7
1.4 Digitalizacija življenjskih procesov	9
2 STORITVE DIGITALNEGA NAROČANJA	10
2.1 Opredelitev digitalnega naročanja in digitalno naročanje v praksi.....	10
2.2 Prednosti digitalnega naročanja	10
3 RAZVOJ NOVE APLIKACIJE NA PRIMERU STORITVE NAROČANJA	11
3.1 Opis problema	11
3.2 Koristi sistema	12
3.3 Konceptualni načrt razvoja aplikacije.....	12
3.3.1 Diagram aktivnosti registracije stranke na termin.....	12
3.3.2 Podatkovna baza	14
3.3.3 Obrazci.....	15
3.3.4 Izvajalci	17
3.4 Možnosti izboljšave	17
SKLEP	18
LITERATURA IN VIRI	19

KAZALO SLIK

Slika 1: Diagram aktivnosti registracije stranke na termin.....	13
Slika 2: Baza podatkov	14
Slika 3: Obrazci v aplikaciji	15
Slika 4: Obrazec za naročilo stranke na termin na primeru frizerskega salona.....	16

UVOD

Dandanes smo na vsakem koraku obdani z novimi tehnološkimi inovacijami in aplikacijami. Za vsak naš problem je ekipa strokovnjakov, ki išče rešitve in odgovore na naše vsakdanje tegobe. Le redko, pa se uporabnik zaveda kako zahtevna je izdelava produktov. Še redkeje pa se srečujemo z zgodbami uspeha. Za vsakim uspehom podjetja kot so Apple, Google, Facebook, Amazon in drugi, obstaja nešteto neuspešnih izdelkov, zaradi katerih so podjetja zaprla svoja vrata. Pomislite na vse izdelke, ki ste jih uporabili v zadnjem letu. Koliko od teh izdelkov imate radi? Koliko jih sovražite? Koliko se jih sploh spomnite?

Trenutno pa se kupci in potencialni uporabniki srečujejo z vse večjo izbiro produktov, ki so si med seboj precej podobni. Na prvi pogled bi celo pomislili, da izdelava aplikacije s pomočjo prave ekipe sploh ni zahtevna. Ustvarjanje produkta ni naključen in enostaven proces, ki poteka od ideje do zasnove in nastanka aplikacije. Gre za kompleksne procese iz različnih področjih, ki jih skupaj imenujemo življenjski cikel upravljanja produkta (angl. Product Lifecycle Management).

Upravljanje življenjskega cikla produkta je proces, ki vključuje ustvarjanje ideje, opis koncepta, poslovne analize, načrtovanje izdelka, informacijsko arhitekturo, ter tehnično izvedbo in na koncu samo izdelavo produkta.

Zaključno strokovno nalogo sem oblikoval s pomočjo povzemanja literature, kjer sem uporabil raznovrstno strokovno literaturo, kot so: znanstveni članki, strokovne knjige in ostala literatura.

Zaključna strokovna naloga je razdeljena na teoretičen in praktičen del. V teoretičnem delu vam bom predstavil vse teoretične podrobnosti, ki vam bodo omogočile lažje razumevanje življenjskega cikla razvoje aplikacije, ter različne modele, ki se uporabljajo pri razvoju aplikacije. Opisal bom tudi trenutni proces naročanja strank na različne storitve, ki v večini še vedno poteka preko telefonskega klica ali fizičnega prihoda v salon/studio.

V praktičnem delu zaključne strokovne naloge, pa vam bom predstavil mojo teoretično programsko rešitev, ki sem si jo zamislil. Podrobneje bo opisan problem starega načina naročanja, prav tako tudi možne koristi, ki bi jih takšna programska rešitev lahko prinesla. Praktični del bo vseboval tudi opis same aplikacije, vključno z opisom podatkovne baze, obrazcev, izvajalcev ter predstavitev končne programske rešitve.

Na koncu dela, pa bo sledil še sklep, v katerem bo na kratko povzeto bistvo zaključne strokovne naloge, ter končne ugotovitve in možnosti izboljšave.

1 APLIKACIJA

1.1 Definicija aplikacije

V zadnjih letih se je mobilna tehnologija korenito spremenila, saj aplikacije in naprave postajajo vse bolj izpopolnjeni operacijski sistemi. Proizvajalci operacijskih sistemov, kot so Apple, Google in Microsoft na svojih sistemih omogočajo in razvijajo ogromno različnih mobilnih aplikacij. Raziskave so ugotovile, da podjetja, ki nimajo vzpostavljenih mobilnih strategij za potrošniške storitve, izgubijo kar dve tretjini poslovnih priložnosti, ki bi jih lahko pridobile, če bi se pridružile trendu ponujanja spletnih storitev (Hoehle, 2015, str. 435-436).

Mobilne naprave in njihove aplikacije uporabnikom zagotavljajo pomembne prednosti v smislu prenosljivosti, zavedanja lokacije in dostopnosti. Nižje cene ter izboljšave strojne in programske opreme pri pametnih telefonih oz. tako imenovanih "ročnih računalnikih", so privedle do izjemne širitve mobilnih in povezanih trgov. Zaradi tega je bilo v zadnjih nekaj letih razvitih ogromno število mobilnih aplikacij. Uporabnost mobilnih naprav in njihovih aplikacij se razlikuje od drugih računalniških sistemov, saj so njihove značilnosti drugačne. Potrebe po programski opremi ročnih računalnikov, kot so - mobilni telefoni, vplivajo na proces razvoja mobilnih aplikacij, saj so le-te vgrajene v telefone med izdelavo ali pa jih stranke namestijo z različnih platform za distribucijo mobilne programske opreme, kot sta Apple App Store in Google Android Market. Uporabniki se odločajo za mobilne aplikacije, ki so se jih zmožni enostavno naučiti in zahtevajo manj časa za izvedbo določene naloge. Glavni cilj aplikacij je olajšati naše vsakodnevne naloge (Nayebi, Desharnais & Abran, 2012, str. 1-2).

Kako pa pravzaprav aplikacijo definiramo?

Preprosto povedano jo definiramo kot vrsto programske opreme, ki uporabniku omogoča izvajanje določenih nalog. Aplikacije, ki poslujejo na namiznih in prenosnih računalnikih imenujemo namizne aplikacije. Tiste aplikacije, ki domujejo na mobilnih napravah pa so si prislužile naziv mobilne aplikacije. Aplikacija bo delovala znotraj operacijskega sistema, dokler je uporabnik ne bo zaprl. V praksi pogosteje v mobilnih napravah pa v ozadju poteka več procesov in s tem hkrati deluje več odprtih aplikacij – kar imenujemo večopravnost (angl. multitasking) (Goodwill Community Foundation, Inc., brez datuma). Beseda 'aplikacija' se najpogosteje nanaša na mobilno aplikacijo ali del programske opreme, ki je nameščena ali uporabljena na uporabnikovem računalniku. Večina aplikacij ima precej specifično in ozko funkcijo. Na primer: aplikacija za dostavo je lahko zasnovana samo za uporabnike, z namenom, da si lahko preko telefona ali računalnika naročajo hrano iz lokalnih restavracij. Posledično to pomeni, da aplikacije ni mogoče uporabljati za širše namene, kot je na primer nakupovanje živil ali rezerviranje svoje najljubše restavracije (Indeed Editorial Team, 2021).

Mobilne aplikacije lahko razdelimo v tri kategorije, in sicer:

- Native oz. domače mobilne aplikacije,
- spletne aplikacije,
- in hibridne mobilne aplikacije.

Nativne oz. domače mobilne aplikacije (angl. native applications) se nanašajo na operacijski sistem aplikacije, ki je specifično zasnovana in razvita za določeno mobilno napravo. Trije vodilni operacijski sistemi so Googlov Android, Applov iOS in Windows Phone. Nativne aplikacije so zelo učinkovite, saj težijo k optimizaciji uporabniške izkušnje. Posledično, ker so te aplikacije razvite za delovanje na samo eni platformi, lahko delujejo veliko bolj učinkovito, hitreje in uporabniku prijazno (Jobe, 2013, str. 28).

Spletna aplikacija je vrsta aplikacije, ki za popolno uporabo potrebuje dostop do interneta. Še pred nekaj leti je bila uporaba izvirnega kompleta za razvoj programske opreme (SDK ali devkit) je bil edini način za ustvarjanje mobilne aplikacije. Vsaka platforma, kot so Android, iOS in Windows, zahteva drugačen SDK, vsaka platforma pa uporablja tudi drugačne programske jezike za kodiranje. Spletne stran ideja o razvoju aplikacije brez kodiranja v Javi ali Objective C se je zdela precej oddaljena. Nedavno, se je okolje pri razvoju mobilnih aplikacij bistveno spremenilo, zlasti s pojavom hibridnih okvirov za mobilne aplikacije (Huynh, Ghimire & Truong, 2017, str. 50).

Hibridne mobilne aplikacije so kombinacija spletnih in domačih aplikacij. V zadnjih letih je hibridni pristop k razvoju mobilnih aplikacij postal nova in obetavna alternativa v razvoju aplikacij. Hibridna aplikacija omogoča razvijalcem, da ustvarijo eno samo mobilno aplikacijo, z uporabo spletnih standardov in njeno dosledno distribucijo po več mobilnih platformah z (minimalnimi ali) ničnimi spremembami. Hibridna aplikacija je v bistvu majhno spletno mesto, napisano v jezikih HTML, CSS in JavaScript. Od običajnih spletnih mest se razlikuje po tem, da deluje le v lupini brskalnika in ima dostop do nativne plasti platforme. Hibridna aplikacija zasnovana in kodirana kot spletno mesto, vendar je nato opremljena z možnostmi za dostop do funkcij izvorne platforme. Rezultat je učinkovit in temelji na spletni tehnologiji, ki deluje in je videti kot nativna aplikacija. Pri hibridnih aplikacijah pa je vredno omeniti, da imajo te aplikacije pogosto nižjo stopnjo učinkovitosti kot domače ali spletne aplikacije (Malavolta, Ruberto & Terragni, 2015).

1.2 Življenjski cikel razvoja aplikacije

Na trgu se vse pogosteje srečujemo z izboljšanimi, strokovnimi in visoko-kakovostnimi modeli življenjskega cikla razvoja produktov. Vedno bolj konkurenčni trg sili podjetja, da razvijajo in dobavljajo produkte višje kakovosti z večjo zmogljivostjo v čim krajšem času in čim nižjimi stroški. Model življenjskega cikla razvoja aplikacije prikazuje pomembne faze razvoja, ki potekajo od zasnove izdelka do njegovega nastanka.

Proces vključuje načrtovanje, oblikovanje, razvoj, testiranje in uvajanje s stalnim vzdrževanjem za učinkovito ustvarjanje in upravljanje aplikacij. Ko se soočite z nalogo izdelave visokokakovostne programske opreme, ki izpolnjuje naročnikova pričakovanja, zahteve, časovni okvir in ocene stroškov, je razumevanje temeljnega procesa za uspešen življenjski cikel razvoja programske opreme ključnega pomena (Gumus, Ertas, Tate & Cicek, 2008, str. 185).

Model razvoja mobilne aplikacije (angl. Mobile Application Development Lifecycle) je sestavljen iz naslednjih korakov (Kaur & Kaur, 2015, str. 4):

1. Faza identifikacije (angl. Identification phase).
2. Faza oblikovanja (angl. Design phase).
3. Faza razvoja (angl. Development phase).
4. Faza oblikovanja prototipa (angl. Prototyping phase).
5. Faza testiranja (angl. Testing phase).
6. Faza nameščanja (angl. Deployment phase).
7. Faza posodabljanja (angl. Maintenance phase).

V nadaljevanju bomo korake predstavili bolj podrobno.

1) Faza identifikacije (Villaumbrosia & Anon, 2017, str. 18): Na začetku imajo naročniki pogosto predstavo kako bo produkt deloval in izgledal, čeprav njihova ideja pogosto ni skladna z praktično izdelavo aplikacije. Pri tem mu razvijalec predstavi ogromno podvprašanj na katere morda ni pripravljen kot so na primer:

»Na katerem operacijskem sistemu želite, da vaša aplikacija deluje?« *»Na kakšni vrsti zaslona pametnega telefona želite, da vaš dizajn deluje?«* *»Kaj je glavna funkcija aplikacije?«* *»Zakaj bi si uporabnik sploh prenesel to aplikacijo?«* *»Kaj je glavna funkcija aplikacije?«,* in *»Kakšno korist bo uporabniku nudila aplikacija?«*

Zato je toliko bolj bistveno, da naročnik in razvijalec najdeta "skupni jezik" in skupaj pripravita tehnično dokumentacijo. Pogosto so podprte s poglobljenimi raziskavami in načrtovanjem. Njegovi pod-koraki so sestavljeni iz (Villaumbrosia & Anon, 2017, str. 19):

- Jasno opredeli idejo, ki jo želi uresničiti.
- Strateško razumevanje podjetja.
- Jasna opredelitev ciljev podjetja.
- Oblikovanje in potrditev priložnostnih hipotez.

2) Faza oblikovanja: Naročnik se odloči za razvoj aplikacije na podlagi vseh opravljenih raziskav kot so npr. uporabniške raziskave in vedenjske raziskave. Faza oblikovanja je razdeljena na dva dela:

- Uporabniško izkušnjo (angl. UX Design)

- Interakcijsko oblikovanje (angl. UI Design)

Uporabniška izkušnja je področje analize vedenja ljudi in zaznavanja izkušenj pri delu s stroji, sistemi in izdelki. Za uspešno uporabniško izkušnjo mora biti produkt (Peruzzini, Grandi & Pellicciari, 2017, str. 806-810):

- Praktičen – Izdelek mora reševati uporabniške težave in zagotoviti, da izpolnjuje potrebe potrošnika.
- Lahek za uporabo – Enostavnost uporabe in možnost učenja sta ključna za ohranitev in pridobitev novih strank.
- Zaželen - Ustvariti moramo čustveno vez z blagovno znamko oz. identiteto produkta, ki pomembno vpliva na celotno uporabniško izkušnjo.
- Dostopen – Prizadevati si moramo, da so čim bolj inkluzivni in je produkt dostopen čim širši publiki, ne glede na telesne ali kognitivne motnje.
- Koristen - Izdelek mora prinesiti vrednost tako uporabniku kot podjetju.

Temelj odličnega izdelka je dobra uporabniška izkušnja, saj gre pri oblikovanju izdelka pogosto za ustvarjanje nove ali izboljševanje obstoječe izkušnje. Oblikovalec uporabniške izkušnje (angl. UX Designer) je glas potrošnika in je zadolžen za raziskavo, informacijsko arhitekturo in oblikovanje prototipa izdelka. Oblikovanje produkta poteka po dveh smernicah. Prva je, da se mora produkt prilagoditi željam potrošnika, drugi pa, da mora izdelek biti ustvarjen na način, da ga uporabnik razume in je lahek za uporabo (Villaumbrosia & Anon, 2017, str. 188-189).

Interakcijsko oblikovanje. V tem koraku se ključne funkcije aplikacije začnejo podrobneje pojavljati in vse naročnikove ideje za aplikacijo dobijo svojo vizualno podobo. V stroki ta korak imenujemo interakcijo oblikovanje (angl. Interaction design). Interakcijsko oblikovanje je praksa oblikovanja interaktivnih digitalnih izdelkov, okolij, sistemov in storitev - pri čemer je treba raziskati, kako lahko uporabnik z njimi komunicira. Običajne področja vključujejo oblikovanje interakcije med človekom in računalnikom ter razvoj programske opreme (Cooper, Reimann & Cronin, 2014, str. 21). Interakcijo oblikovanje si pomaga z informacijsko arhitekturo ustvariti okvir za digitalni izdelek, storitev, spletno mesto ali aplikacijo. Produkt bo uspešen, če ga bo ga uporabnik sposoben v celoti razumeti ali se ga naučil uporabljati z najmanjšim možnim naporom in brez zmede. Naročnik mora v tem koraku ustvariti grob priročnik za izgled aplikacije in morda uporabiti pomoč tehničnega pisca. To bo omogočalo vpogled v funkcionalnost aplikacije, ali pa bo pripomoglo, da se opazijo kakšne pomanjkljivosti ali napake, ki so predhodno morda bile spregledane. Ustrezno dokumentiranje različnih vidikov aplikacije, ne pomaga le naročniku, ampak ustvari tudi *wireframe* končne aplikacije (Tidwell, 2010, str. 27).

Wireframes so žične slike najboljši način za predstavitev povezav med različnimi zasloni, vključno z obešenimi televizorji z nosilcem za televizorje s polnim gibanjem, in za določitev, kako bo spletno mesto delovalo s praktičnega vidika. Na podlagi informacij, zbranih med

raziskavo, in odločitev, sprejetih med oblikovanjem hierarhije, bo ocenjevalec vplivov na okolje narisal nekatere ključne zaslone, da bi prikazal, kako bo uporabnik komuniciral z razpoložljivimi informacijami (Rosenfeld & Morville, 2002, str. 90-95).

3) Faza razvoja (angl. Software Development): se nanaša na vrsto dejavnosti računalništva, namenjenih procesu ustvarjanja, načrtovanja, nameščanja in podpore programske opreme. Programska oprema je skupek navodil ali programov, ki računalniku narekujejo, kaj naj naredi. Je neodvisen od strojne opreme in omogoča programiranje računalnikov. Poznamo tri osnovne vrste: *Sistemska programska oprema* za zagotavljanje osnovnih funkcij, kot so operacijski sistemi, upravljanje diskov, pripomočki, upravljanje strojne opreme in druge operativne potrebe. *Programska oprema*, ki programerjem omogoča uporabo orodij, kot so urejevalniki besedil, prevajalniki, povezovalniki in druga orodja za ustvarjanje kode. *Aplikativna programska oprema* (aplikacije ali aplikacije), ki uporabnikom pomaga pri opravljanju nalog.

V tem koraku se začetni prototipi pretvorijo v dejansko aplikacijo. Ekipa programerjev, mora upoštevati vse delujoče prototipe, funkcionalnost aplikacije, skice, in diagrame poteka dela. Glavna naloga je zagotoviti, da je uporabniška izkušnja pri uporabi aplikacije čim bolj 'uporabniku prijazna' in lahka za uporabo. Back-end razvoj je zelo kompleksna in funkcionalna komponenta, ki dejansko zagotavlja celotno funkcionalnost sistema. Sam izraz back-end se nanaša na strežniško stran aplikacije in je zadolžen za vse operacije, izračune, in nenazadnje tudi zanesljivost aplikacije (Software, 2020).

4) Faza oblikovanja prototipa: Ta faza je namenjena prepoznavanju težav v razvoju. Prototipi se testirajo in pošljejo strankam, uporabnikom z namenom, da pridobijo povratne informacije. Po povratnih informacijah se zahtevane spremembe izvedejo preko razvojne faze. Ko je drugi prototip pripravljen, ga vključimo v prvi prototip, preizkusimo in pošljemo stranki. Razvojne, prototipne in testne faze se ponavljajo, dokler ni pripravljen končni prototip. Glavni cilj je prikazati izvedljivost prototipa mobilne aplikacije za izboljšanje aplikacije v času samega razvoja. Oblikovanje prototipa vsebuje: namen načrtovanja in pojasnila; poročanje o kršitvah; poročanje o težavah; informacije o produktivnosti; naloge za spremembo; dnevna poročila, (Nourbakhsh, Zin, Irizarry, Zolfagharian & Gheisari, 2012, str. 474-477).

5) Faza testiranja: Izraz mobilno testiranje se nanaša na različne vrste testiranja, kot so testiranje nativnih mobilnih aplikacij, testiranje mobilnih naprav in testiranje mobilnih spletnih aplikacij. S testiranjem mobilnih aplikacij označujemo "dejavnosti testiranja nativnih in spletnih aplikacij na mobilnih napravah z uporabo natančno opredeljenih metod in orodij za testiranje programske opreme za zagotavljanje kakovosti funkcij, obnašanja, delovanja in kakovosti storitev ter lastnosti, kot so mobilnost, uporabnost, povezljivost, varnost in zasebnost." Testiranje mobilnih aplikacij se od običajnega testiranja programske opreme razlikuje po več posebnih zahtevah. Prvič, mobilne

aplikacije morajo pravilno delovati vedno in povsod. Ker so mobilne storitve pogosto razvite za določeno število izbranih naprav, morajo aplikacije pravilno delovati na vseh napravah. Platforme, ki imajo na primer različne operacijske sisteme, velikosti zaslonov, računalniške vire in življenjsko dobo baterije (Gao, Bai, Tsai & Uehara, 2014, str. 46).

Faza testiranja se vključuje (Gao, Bai, Tsai & Uehara, 2014, str. 47):

- Testiranje funkcionalno in obnašanje aplikacije (dejavnosti, s katerimi se preverjajo funkcije storitev, mobilni spletni vmesniki, uporabniški vmesniki in obnašanje sistema).
- Testiranje obremenitve sistema, zmogljivosti, zanesljivosti/razpoložljivosti.
- Testiranje operacijskega sistema v povezavi z različnimi napravami in platformami.
- Testiranje varnosti in zasebnosti, s katerimi se preverja varnost naprave, vdor v sistem.
- Testiranje mobilnosti, ki potrjujejo lokacijske funkcije, uporabniški profil, sistemske podatke in uporabniške podatke.
- Testiranje združljivosti in povezljivosti platforme.

6) Faza nameščanja: Po končanem testiranju in pridobitvi končnih povratnih informacij od stranke je aplikacija pripravljena za produkcijo (Kaur & Kaur, 2015, str. 4).

7) Faza posodabljanja: Uporabniki zberejo povratne informacije in na podlagi le-teh se izvedejo potrebne spremembe v obliki popravkov ali izboljšav. Trgovine z aplikacijami, kot sta Google Play in Apple Store, uporabnikom že omogočajo, da posredujejo povratne informacije o aplikacijah z objavljanjem komentarjev in ocenjevanjem z zvezdicami. Te platforme so uporabno elektronsko sredstvo, s katerim lahko razvijalci aplikacij in uporabniki produktivno izmenjujejo informacije o aplikacijah. Prejšnje raziskave so pokazale, da povratne informacije uporabnikov vsebujejo scenarije uporabe, poročila o napakah in zahteve po funkcijah, ki lahko razvijalcem aplikacij pomagajo pri opravljanju nalog vzdrževanja in razvoja programske opreme. Razvijalci so v ta namen začeli uporabljati tako imenovani *ARminer* pristop, ki razvijalcem pomaga odkriti najbolj informativne ocene uporabnikov. Najpogosteje uporabljajo analizo besedila in strojno učenje za izločanje ne-informativnih mnenj (Panichella in drugi, 2015, str. 281).

1.3 Predstavitev različnih modelov razvoja aplikacij

V nadaljevanju bomo obravnavali sledeče modele:

Model slapa ali "waterfall" model je linearni sekvenčni tok, ki se najpogosteje uporablja pri projektih, ki imajo določen nabor zahtev. Proces modela, ki ga odraža že njegovo ime, teče vztrajno navzdol skozi faze implementacije programske opreme. Model temelji na zaključku predhodne faze, da se nato začne naslednja faza. Sam model ne podpira vračanja na

predhodno zaključene faze, zato je treba model slapa uporabiti pri projektih, ki ne predvidevajo nepredvidenih sprememb sredi razvojnega cikla. Ta pristop razvoja je bil eden najzgodnejših modelov, ki se je uporabljal za razvoj programske opreme. Ena izmed prednosti tega modela je, da je zelo enostaven za uporabo. Primeren je za upravljanje življenjskih ciklov manjših projektov, kjer so zahteve dokončno določene vnaprej, ga je pa veliko težje implementirati v kompleksnejše projekte, kjer obstaja velika možnost sprememb ali improviziranja v razvojni fazi programske opreme (Bassil, 2012, str. 1-2).

V-model je razširjena verzija modela slapa, ki prav tako deluje kot sekvenčni tok, vendar pa se življenjski cikel razvoja programske opreme po kodiranju ne premika samo linearno navzdol, ampak se upogne tudi nazaj navzgor. Za vsako fazo v zaporedju navzdol, obstaja ustrezna faza testiranja v naslednjem zaporednem koraku navzgor. Prednost V-modela v primerjavi z modelom slapa je, da le-ta lahko prinese večje možnosti za uspeh končnega izdelka, zaradi načrtovanega testiranja v razvojni fazi. Je pa podobno kot metoda slapa sama po sebi zelo toga, zato ni idealna za razvoj aplikacije ali systemske programske opreme, ki zahteva nepredvidene spremembe skozi celoten življenjski cikel (Mathur & Malik, 2010).

Inkrementalni model (angl. Incremental model) je bil zasnovan, zaradi pomanjkljivosti modela slapa. Model razdeli razvojne faze na majhne stopnje in lahko razvijalcem programske opreme omogoči, da izkoristijo nova znanja in vpogleda, iz prejšnjih razvojnih stopenj. Je odlična rešitev za projekte, ki morajo tekom svojega razvoja uvajati spremembe na prejšnjih korakih, z njim pa je mogoče hitreje zaznati težave pri razvoju programske opreme in omogoča boljše načrtovanje upravljanja življenjskega cikla. Slabost, ki se pri modelu lahko pojavi pa je večja potreba po strateškem načrtovanju in pripravi dokumentacije projekta, kar pa običajno zahteva tudi več virov, osebja in denarja za izvedbo (Ruparelia, 2010, str. 9).

RAD model (angl. Rapid application development) oziroma model hitrega razvoja aplikacij, je modificiran inkrementalni model. Pri izvajanju tega modela, se sočasno razvija več komponent, katere so razdeljene na več manjših projektov, ki se pozneje združijo v celoto delujočega prototipa. Model skrajša razvojni čas in omogoča več povratnih informacij uporabnikov ali naročnikov v celotnem razvoju programske opreme. Lahko pa deluje tudi kot zelo omejen, saj lahko pride tudi do težav ob končnem združevanju komponent, kar pa lahko povzroči nepričakovane zaplete pri končnem združevanju delov v celotno aplikacijo (Ruparelia, 2010, str. 12).

Agilen model temelji na prilagodljivih metodah razvoja programske opreme, medtem ko tradicionalni modeli kot so na primer model slapa temeljijo na napovednem pristopu. V tradicionalnih modelih SDLC ekipe delajo s podrobnim načrtom in imajo popoln seznam značilnosti in nalog, ki jih je treba opraviti v naslednjih nekaj mesecih ali v celotnem življenjskem ciklu izdelka. Agilni model se močno nanaša na sodelovanje med več-funkcionalnimi ekipami in naročnikom/uporabnikom. Omenjen model strmi k zmanjševanju časa pridobitve posameznih systemskih funkcij, kar sicer zahteva ogromno komunikacije

med proizvajalci in naročniki, ampak s tem lažje zagotovi jasno usmerjenost projekta. Slabost, ki se pogosto pojavlja je le-ta, da se zanaša na interakcijo s končnimi uporabniki, pri čemer lahko slaba komunikacije vpliva na odmik iz načrtovane poti (Stoica, Mircea & Ghilic-Micu, 2013, str. 71-72).

Spiralni model poskuša odpraviti dve glavni težavi modela slapa (Ruparelia, 2010, str. 10-11):

1. Dve fazi načrtovanja sta lahko v nekaterih primerih nepotrebni
2. Pristop od zgoraj navzdol je treba ublažiti s korakom vnaprej, da se upošteva ponovna uporabnost programske opreme ali zgodnjega prepoznavanja tveganj, vprašanj.

Model vsebuje štiri faze: (1) določitev ciljev in identifikacijo; (2) ocenjevanje alternativ, ter ugotavljanje in odpravljanje tveganj; (3) razvoj in testiranje; (4) načrtovanje iteracije. Z napredovanjem vsakega cikla se izdela prototip, preverijo zahteve in potrdi se testiranje. Pred tem se tveganja opredelijo in analizirajo. S tem omogoča natančnejše ocene proračuna in urnika, saj so ovire odkrite zgodaj v projektu. Ker pa je model zelo prilagojen, je ponovna uporaba postopka lahko zmedena, prav tako pa zahteva, da ekipo sestavljajo člani, ki so dobro seznanjeni z oceno tveganja (Aimicheva, Kopeyev, Ordabayeva, Tokzhigitova & Akimova, 2020, str. 1878).

1.4 Digitalizacija življenjskih procesov

Priča smo fenomenu kjer se digitalen in fizičen svet vedno bolj prepletata. Digitalizacija se vse bolj napreduje in zajema fizično sfero, sočasno pa omogoča virtualno digitalizacijo vsega, kar je mogoče digitalizirati. Ker pa so vsakdanje stvari opremljene z zmožnostjo zaznavanja svojega okolja - odražajo lokacijo, uporabljajo zgodovino, status in delijo informacije z drugimi predmeti/subjekti -, se naše doživljanje resničnega življenja in nas samih spreminja na veliko različnih načinov (Fors, 2013, str. 45).

Najenostavneje povedano beseda digitalizacija pomeni pretvorbo analognih informacij v digitalno obliko. Zgodovino digitalizacije lahko povežemo z samim začetkom dobe računalnikov. Z povečevanjem zmogljivosti računalnikov, sta naraščali tudi hitrost in uspešnost digitalizacije. Proces pa je dobil nov pospešek s prihodom dobe interneta. Digitalizacija se uporablja že skoraj v prav vseh vidikih človeških življenj. Končna evolucija procesa digitalizacije je vrhunec digitalizacije vsakodnevnih človeških opravil. Internet stvari (angl. Internet of Things - IoT) ni več znanstvena fantastika ampak je zelo resničen koncept, kjer se vsak vidik človeškega življenja spremeni v 1 in 0. Procesorska moč eksponentno narašča, mikro-procesni čipi postajajo vse manjši, komunikacijska omrežja pa znova in znova podirajo nove rekorde hitrosti in pasovne širine. Tehnologija zaznavanja postavlja nove temelje, te senzorje pa vgrajujemo v vse, od oblacil (za uravnavanje temperature) do tekaških copat (za oceno uspešnosti športnika) do sistemov za avtomatizacijo doma (za ustvarjanje tako imenovanega pametnega doma). Vse to omogoča

ustvarjanje pametnejše in bolj povezane družbe. Fenomenalen vzpon družbenih omrežij je dokaz človeškega zavzemanja za digitalno življenje (Borderless Technology Corp., 2017).

2 STORITVE DIGITALNEGA NAROČANJA

2.1 Opredelitev digitalnega naročanja in digitalno naročanje v praksi

Na splošno vsaka organizacija ali poslovna panoga zahteva, da se oseba digitalno naroči na storitve, ki jih je treba potrditi ali preveriti. V današnji družbi so posamezniki običajno na poti, zato se za naročanja hitro dogovorijo, spremenijo in potrdijo, kar zahteva potrpežljivost, čas in stroške. Poleg tega je običajno časovni zamik med zahtevo za termin in odobritvijo termina. Eno od takšnih področij, kjer se ukvarjajo z digitalnim naročanjem je medicina (Akshay, Kumar, Alagappan & Gnanavel, 2019, str. 1-2).

Kjer mora pacient ali uporabnik rezervirati termin pri družinskem zdravniku, ki lahko nato zahteva, da se pacient rezervira pri specialistu, v bolnišnici, krvnem laboratoriju, rentgenu ali drugi ustanovi za skeniranje. Obstajajo različni poskusi za izboljšanje tega vprašanja razporejanja. Enak scenarij velja tudi za digitalno naročanje v salonu, vodje dogodkov, gospodinjske storitve itd. Cilj takšnih sistemov je zagotoviti izboljšano spletno storitev rezervacij. Nadaljnji cilj sistema bi bil zagotoviti enotno centralizirano storitev rezervacij ne le za bolnike, temveč tudi za uporabnike, ki potrebujejo mizarje, vodovodarje, arhitekto (Akshay, Kumar, Alagappan & Gnanavel, 2019, str. 1).

Danes je splošno priznano, da mora dobro načrtovan zdravstveni sistem zagotoviti pravočasen in enostaven dostop do zdravstvenih ustanov za vse bolnike. Digitalno načrtovanje terminov lahko izboljša produktivnost dragega osebja in zdravstvenih virov ustanov, hkrati pa skrajša čakalne dobe pacientov. Namen digitalnega naročanja na zdravstvene preglede je zgraditi sistem naročanja, ki optimizira določen standard kakovosti v aplikaciji zdravstvenega varstva. Primarna funkcija sistema je čim bolj zmanjšati čakalne dobe pacientov v bolnišnicah in povečati njihovo zadovoljstvo (Ala & Chen, 2022, str. 1).

Pandemija COVID-19 je predvsem še bolj spodbudila sprejetje digitalnih zdravstvenih tehnologij za čim večjo dostopnost primarne zdravstvene oskrbe. Predhodne študije o sistemih za digitalno naročanje na zdravstvene preglede se osredotočajo na zaznane rezultate, kot je na primer zadovoljstvo pacientov, ali pa na tehnično perspektivo, ki se je osredotočala na optimizacijo zdravstvenih algoritmov za načrtovanje (Ala & Chen, 2022, str. 1).

2.2 Prednosti digitalnega naročanja

Pri vsaki novi tehnologiji obstaja obdobje, kjer se družba na tehnologijo prilagodi, preden se uporaba nove tehnologije razširi. Ljudje potrebujejo čas, da vidijo vrednost in potencial

tehnološke rešitve, ki pa se sčasoma normalizira. S programsko opremo za naročanje na digitalno naročanje se je ta trend ohranil. Trenutno je industrija v obdobju velike rasti (QLess, Inc., brez datuma).

Forbes je digitalno naročanje uvrstil med glavne vzhajajoče trende v panogah, ki temeljijo na naročanju terminov. Glede na Forbesovo poročilo se je industrija hitro razširila tudi na področje zdravstvenega varstva, pri čemer bolnišnice in zdravniki strmijo, da se prilagodijo potrebam mlajše generacije in njihovi željam po vedno večji digitalizaciji. Vedno pogosteje se pojavljajo rešitve za razporejanje sestankov v oblaku postajajo standardi za programsko opremo za razporejanje sestankov in upravljanje odnosov s strankami (Asli, 2019).

Vsakič, ko inovacija izboljša delovanje podjetij po dostopni ceni, ne mine dolgo, preden postane ta inovacija industrijski standard. Programska oprema za digitalno naročanje vsako leto vztrajno raste, pri čemer *Allied Market Research* predvideva, da bo velikost trga med letoma 2019 in 2026 narastla iz 205 milijonov ameriških dolarjev na 546 milijonov dolarjev. Vse več podjetji se namreč odloča za prehod iz ročnega upravljanja koledarjev na programska orodja, ki namesto vas upravljajo vse zunanje koledarje. S spletnimi rezervacijami so podjetja bolj organizirana, in dosegajo zastavljene urnike, enako pa velja tudi za stranke (QLess, Inc., brez datuma).

Po podatkih *FinancesOnline*, so samodejni opomniki za digitalno naročanje iz spletne programske opreme za načrtovanje zmanjšali neprihod strank za kar 29%. V številnih panogah se vsi vidiki poslovanja pri katerih je to možno implementirati usmerjajo k digitalizaciji procesa. Vedno več strank -sploh pripadniki mlajših generacij- se vedno bolj nagibajo k udobju, ki ga digitalne storitve ponujajo. Investiranje v digitalizacijo je v zadnjih nekaj letih sunkovito narastla. Po podatkih *Statista* se je svetovna uporaba tehnologije za digitalne preobrazbe povečala leta 2017 za 960 milijonov dolarjev, predvidevajo, da bo do leta 2024 narastla na 2,39 milijarde dolarjev. Za podjetja je bistveno, da se posodablajo, da bodo ustrezala potrebam svojih strank, digitalnemu naročanju in bolj kakovostnem upravljanju čakalnih vrst (QLess, Inc., brez datuma).

3 RAZVOJ NOVE APLIKACIJE NA PRIMERU STORITVE NAROČANJA

V tem poglavju se bomo osredotočili na opis programske rešitve, ki temelji na primeru storitev digitalnega naročanja.

3.1 Opis problema

Uporabniki storitev, se še vedno v veliki meri na storitve naročajo preko telefonskega klica ali preko elektronske pošte. To velja za skoraj vse obstoječe storitve, ki segajo od naročanja

na zdravniški pregled do bolj enostavnih storitve kot je na primer naročanje na frizerske storitve.

Naročanje preko telefona je lahko neprijetna - včasih celo časovno obremenjujoča. Z vidika stranke/ uporabnika, ki se na storitev naroča preko telefona lahko nastane problem pri komunikaciji, ki pa se povezuje z organizacijskim sistemom, ki ga podjetje ima.

Predstavljajte si naslednji scenarij: *»Jaz kot stranka kličem, da se naročim na željen termin, ki bi meni ustrezal. Žal nimam nobenega pregleda kateri termini so prosti in kateri zasedeni. Z ponudnikom ugotoviva, da morama s skupnimi močmi preko telefona najti prvi prost termin – kar je časovno zamudno in bi lahko bilo rešeno z boljšo preglednostjo prostih terminov. Potem pride še tukaj do problema pravilnega beleženja in načrtovanja storitev, kar lahko v veliki poplavi informacij se izgubi.«*

Problem lahko nastane tudi, ko stranka pokliče v studio/salon. V primeru, da storitev nima zaposlene nekoga, ki bi urejal administrativne zadeve se mora nekdo od zaposlenih javiti na zaposlenih na telefon. Posledično to pomeni, da mora prenehati opravljati svoje trenutno delo, da se javi na telefon. V primeru, da se ne javi na telefon tvega izgubo stranke. V obeh primerih je lahko nenamerno povzročitelj nezadovoljstva stranke.

Glavni problem naročanja preko elektronske pošte je, da se jo lahko zelo enostavno spregleda, kar pomeni, da izgubi stranko še preden pride do naročanja.

3.2 Koristi sistema

S tem namenom, smo si zamislil teoretično programsko rešitev, ki bo reševala zgoraj omenjene probleme. Naš cilj je predstaviti aplikacijo, ki bi izboljšala uporabniško izkušnjo stranke, pripomogla k boljši organizaciji sistema in omogočila, da bo proces digitalnega naročanja potekal hitreje in enostavneje.

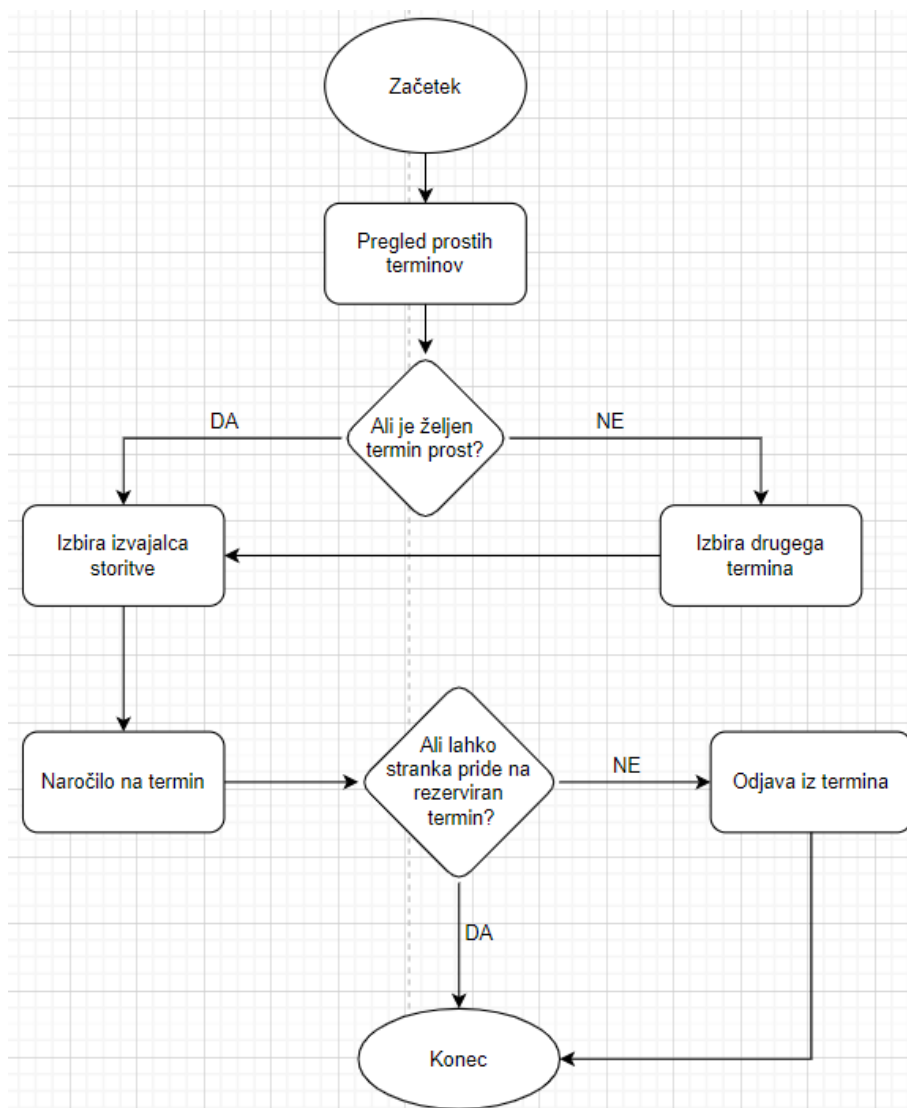
To bi omogočili tako, da bi v aplikaciji imeli seznam, na katerem bi bili prikazani vsi prosti termini, ki še niso zasedeni. Stranka bi nato izbrala termin, ki ji najbolj odgovarja na podlagi ponujenega seznama. S tem bi zmanjšali obremenjenost zaposlenih v storitvenih dejavnostih.

3.3 Konceptualni načrt razvoja aplikacije

3.3.1 Diagram aktivnosti registracije stranke na termin

Aplikacija ima več diagramov aktivnosti, ki so neposredno vezani na vrsto uporabnika, ki aplikacijo uporablja. Diagram aktivnosti registracije uporabnika prikazuje na spodnji sliki pot uporabnika, ki se želi naročiti na termin.

Slika 1: Diagram aktivnosti registracije stranke na termin



Vir: lastno delo.

Prvi korak: Proces izbire termina

Iz diagrama lahko razberemo, da je prvi korak za uporabnika, da izbere prost termin med ponujenimi možnostmi, ki jih je storitev razpisala. V primeru, da stranka vidi, da je željen termin prost, le-tega izbere in nadaljuje na naslednji korak v procesu. V obratni situaciji, kjer stranka ugotovi, da je njen željen termin zaseden lahko izbere eno izmed prostih možnosti in nadaljuje na naslednji korak v procesu. Če se zgodi, da stranka ugotovi, da ji noben termin ne ustreza, potem zaključi svojo pot v procesu.

Drugi korak: Proces izbire izvajalca storitve

Stranka bo v naslednjem koraku lahko izbrala izvajalca storitve, ki bodo razpisani skupaj s prostimi termini. Po izbiri izvajalca bo uporabnica lahko zaključila naročilo.

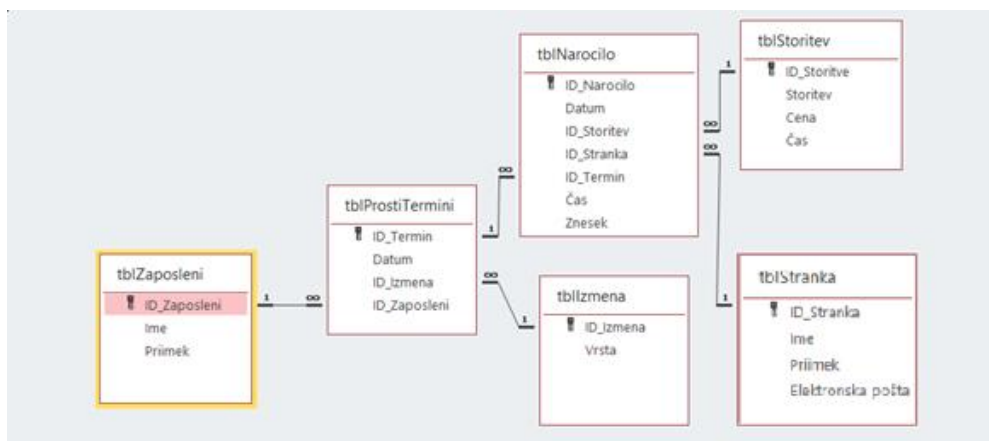
Tretji korak: Aplikacija pošlje opomnik

Aplikacija bo poslala opozorilo par dni pred izbranim terminom, kjer bo ponovno potrdila svoj termin ali pa bo imela možnost pravočasno se odjaviti iz izbranega termina, če bi ugotovila da se le-tega ne bi mogla več udeležiti.

3.3.2 Podatkovna baza

Spodnja slika prikazuje bazo podatkov, ki smo jo naredili za izbrano programsko rešitev je sestavljena iz šestih različnih tabel, in sicer tabela zaposlenih, tabela prostih terminov, tabela izmen, tabela naročil, tabela strank in tabela storitev.

Slika 2: Baza podatkov



Vir: lastno delo.

Tabela zaposlenih je namenjena za hranjenje podatkov o zaposlenih, ki delajo v določeni storitveni dejavnosti. Ta tabela je pomembna, saj omogoča strankam, da se sami odločimo kateri od zaposlenih bo opravljal željeno storitev, saj ima vsaka stranka predvideno svoje preference. Stranka lahko z možnostjo izbire tako dobi željen rezultat opravljene storitve. dobi najboljši možen rezultat. S tem si zagotovimo, če bo stranka zadovoljna z storitvijo se bo vedno znova vračala. Glavni ključ v tabeli je `ID_Zaposleni`.

Tabela prostih terminov vsebuje vse proste termine, med katerimi bodo lahko stranke izbirale. Vsakemu terminu je dodeljen poseben ID (`ID_Termin`), ki je tudi glavni ključ v tabeli. Tabela vsebuje tudi datum prostih terminov med katerimi lahko stranka izbira. `ID_Izmena` in `ID_Zaposleni` pa v tabeli predstavljata tuje ključe in omogočajo povezavo z ostalimi tabelami.

Tabela izmena je namenjena beleženju prostih terminov v dopoldanskem času ali popoldanskem. Uporabnik izbere eno izmed ponujenih, ki mu bolj ustreza. Glavni ključ v tabeli je `ID_Izmena`.

Tabela stranka vsebuje podatke, ki jih stranka vnese ob registraciji računa ob uporabi aplikacije. Glavni ključ tabele je ID_Stranka, saj ima vsaka stranka svojo specifično identifikacijsko številko. Vsebuje tudi ime in priimek stranke, ki se pojavi na naročilu na termin. Elektronski naslov pa je beležen, saj je nanj vezan račun, ki ga uporabnik uporabi pri prijavi v aplikacijo.

Tabela storitev vsebuje vse podatke, ki so vezane na vse vrste storitev, ki jih določena podjetje ponuja. V stolpcu storitev bodo navedene vse različne storitve kot so na primer v našem primeru frizerskega salona: moško striženje, žensko striženje, barvanje las. V stolpcu cena pa so vnesene v naprej določene cene za različne storitve. V stolpcu čas pa je naveden okvirni čas za izvedbo določene storitve. Glavni ključ tabele je ID_Storitev.

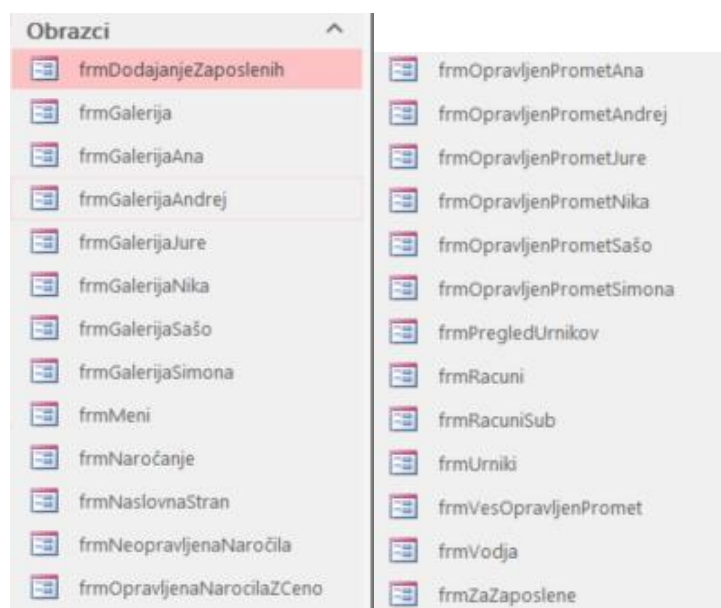
Tabela naročilo je najpomembnejša tabela v celotni bazi podatkov, saj vsebuje vse potrebne podatke, za beleženje in spremljanje trenutnih naročil. Glavni ključ tabele je ID_Naročilo, saj ima vsako naročilo svojo specifično identifikacijsko številko, ki se nikoli ne ponovi.

Polje datum beleži datume rezerviranih terminov. Čas in znesek sta podatka, ki se pridobita iz tabele storitev, in se ob izbiri željene storitve avtomatsko izpolnita. Tabela vsebuje tudi tri tuje ključe, in sicer ID_Storitev, ID_Stranka in ID_Termin, in omogoča povezovanje in črpanje podatkov iz ostalih tabel v bazi podatkov.

3.3.3 Obrazci

Naša aplikacija vsebuje šestindvajset različnih obrazcev, ki so prikazani na spodnji sliki, ki jih lahko razdelimo glede na vrsto uporabnikov, ki lahko do njih dostopajo.

Slika 3: Obrazci v aplikaciji



Vir: lastno delo.

Vsi uporabniki lahko dostopajo do obrazcev galerije, v kateri se nahajajo slike opravljenih storitev v primeru takšne dejavnosti (npr. frizerstvo – slike končnih izdelkov storitev), uporabniki (stranke) pa lahko tudi sami objavijo fotografije, če se za to odločijo. Vsi uporabniki imajo tudi možnost dostopanja do obrazcev meni in naročanje, saj se iz obrazca meni lahko dostopa do vseh ostalih obrazcev, obrazec naročanje pa je namenjen rezervaciji termina na določeno storitev.

Obrazci, do katerih lahko dostopajo samo zaposleni je obrazec za zaposlene, ki omogoča dostop do ostalih obrazcev in pregled urnika zaposlenih. Vsi imajo dostop do obrazca računi, kjer se beležijo do sedaj opravljeni računi storitev, namenjen pa je tudi za izdajo računov. Zadnji obrazec do katerega lahko dostopajo samo zaposleni pa je obrazec neopravljenih naročil, na katerem se beležijo prihajajoča naročila.

Vodja zaposlenih edina oseba v podjetju, ki ima dostop do obrazca dodajanja zaposlenih. Ta funkcija mu omogoča dodajanje novih zaposlenih in odstranjevanje ne-aktivnih zaposlenih iz aplikacije. Edini lahko dostopa tudi do podatkov o opravljenem prometu. Ta funkcija poenostavi pregled nad opravljenim delom zaposlenih in mu lahko pomaga pri dodelitvi stimulacije. Ima pa tudi dostop in možnost urejanja obrazcev za urnik, v katerem lahko dodaja nove urnike, ter spreminja že obstoječe urnike v primeru bolezni.

Slika 4: Obrazec za naročilo stranke na termin na primeru frizerskega salona

The screenshot shows a booking form with the following fields and values:

- ID_Narocilc: 74
- Storitev: Kodranje
- Ime: Jana
- Primek: Novak
- Elektronska pošta: jana.novak@gmail.
- Zaposleni: 60
- Datum: 29. 07. 2022

The service dropdown menu is open, showing the following items:

Striženje kratk	18,00 €	1
Striženje dolg	36,00 €	1
Barvanje las	30,00 €	1
Kodranje	40,00 €	1
Prameni cela	59,00 €	1
Prameni pol c	40,00 €	1
Otroško striže	9,00 €	1

Below the form is a table showing the employee schedule for the selected date (29. 07. 2022):

58	29. 07. 2022	Dop	Jure	Pozne
59	29. 07. 2022	Pop	Nika	Vučko
60	29. 07. 2022	Dop	Simona	Borštinar
61	29. 07. 2022	Pop	Ana	Kušter

At the bottom of the form are two buttons: "Nazaj na MENI" and "Pošlji naročilo".

Vir: lastno delo.

Glavni obrazec za stranko, ki se želi prijaviti na termin je obrazec za oddajo naročila in je prikazan na zgornji sliki. Pri prvem polju stranka iz spustnega seznama izbere željeno storitev, ki poleg naziva storitve vsebuje tudi znesek. Stranka na izbrano storitev klikne, spustni seznam se samodejno zapre, izbrana storitev pa se samodejno zabeleži. Nato vnese

svoje ime in priimek, svojo elektronsko pošto, na katero bo prejel tudi potrdilo o naročilu na izbrano storitev.

V polju zaposleni si stranka, ki se prijavlja na storitev izbere izvajalca storitve, zraven pa je zapisan delavnik zaposlenih. Datum naročila se samodejno izpolni ob izbiri izvajalca storitve. Takrat, ko stranka izpolni vsa potrebna polja pritisne na gumb »Pošlji naročilo«. Če so vsi potrebni podatki pravilno izpolnjeni bo uporabnik dobil potrditveno: »Naročilo je bilo uspešno oddano« v primeru, da pa je kakšno polje prazno ali pa neustrezno izpolnjeno pa dobi stranka obvestilo »Naročilo je nemogoče oddati! RAZLOG: manjkajoči podatki!«.

3.3.4 Izvajalci

Aplikacija ima tri različne vrste uporabnikov: stranke, zaposlene in vodje zaposlenih. Dostopnost aplikacije bi bila različna odvisno od vrste uporabnikov, ki bi aplikacijo uporabljali.

Stranke bi v aplikaciji imele možnost pregleda prostih terminov, ki bi bili vidni v spustnem seznamu. Imeli bi tudi možnost objave na že prijavljen termin v primeru, da se ga ne bi mogli udeležiti. Prav tako imajo možnost oddati mnenje, sliko svoje frizure in oceno svojega zadovoljstva z opravljeno storitvijo.

Zaposleni v organizaciji bi imeli poleg istih funkcij kot stranke, možnost tudi dodajanja lastnih slik izdelkov (v primeru takšne storitve). Prav tako, lahko izdajajo račune, imajo pregled na njihovimi mesečni prihodki preko izdanih računov.

Vodje bi imele možnost opravljanja istih funkcij kot ga imajo stranke in zaposleni. Njihov nabor funkcij bi vključeval še dodajanje novih zaposlenih, odstranjevanje zaposlenih in delo z urniki zaposlenih.

3.4 Možnosti izboljšave

Aplikacija bi doprinesla ogromno koristi, ampak nam je zaradi obsega naše naloge še vedno ostal prostor za izboljšavo. Pomanjkljivost takšne aplikacije je, da je le-ta specializirana za eno organizacijo in bi bilo potrebno razviti precej logistike - kako bi lahko njeno delovanje aplicirali na ostala področja.

To bi se lahko rešilo na način, da bi za naročanje na termine obstajala ena aplikacija, za organizacijo, ki bi na te načine sprejemala naročila pa druga aplikacija, ki je povezana z aplikacijo za naročanje. Tako bi lahko v aplikaciji za naročanje bile vse organizacije, na katerih bi bilo možno rezervirati termin, naročnik termina pa bi si potem samo izbral organizacijo, pri kateri želi rezervirati termin.

Prav tako je ena izmed izboljšav, ki bi jo bilo potrebno nujno narediti, če bi želeli dejansko razviti takšno aplikacijo bi morali uporabiti program, ki je namenjen razvoju programskih rešitev. Trenutna aplikacija je narejena v programu Microsoft Access, kar pa ni pogosto uporabljeno programsko orodje za razvoj programske opreme. Obstaja kar nekaj boljših platform, za izgradnjo aplikacij, kot so na primer Quixy, Zoho Creator in Appery.io, saj omogočajo dejansko izgradnjo aplikacij za android in iOS.

Prav tako, bi bilo potrebno vizualno urediti celotno programsko rešitev, kot sem omenil je bila narejena z pomočjo Microsoft Accessa - ki pa je precej omejen glede možnosti za izboljšavo izgleda.

Končna programska rešitev bi zahtevala veliko dodatnih funkcij, da bi lahko uspela. Ena izmed popularnih funkcij, ki bi lahko bolje prikazala zadovoljstvo strank določene je: možnost ocenjevanja in dodajanja mnenj.

SKLEP

Na svetovnem trgu se vedno pogosteje srečujemo z izboljšanimi, učinkovitimi in visokokakovostnimi modeli življenjskega cikla razvoja produktov. Vedno bolj konkurenčni trg sili podjetja, da razvijajo in dobavljajo izdelke višje kakovosti, z večjo zmogljivostjo, v čim krajšem možnem času in nižjimi stroški. Kar pa ni merilo uspeha aplikacije, ki se pojavi na trgu.

Za uspeh aplikacije dandanes stoji ekipa strokovnjakov, ki pokrivajo področja programiranja, trženja, poslovnega analiziranja, upravljanja produkta in še več, da se zagotovi uspeh produkta. Gre za porazdeljen sklop vlog in povezanih odgovornosti, ki pokrivajo vse vidike procesa razvoja izdelkov in s tem zagotovijo uspešnost produkta.

Pri vsaki novi tehnologiji obstaja obdobje, kjer imajo ljudje določeno prilagoditveno obdobje, preden se uporaba tehnologije razširi. Ljudje potrebujejo čas, da vidijo koristnost in uporabnost tehnološke rešitve, ki se le-ta sčasoma normalizira. S tem namenom smo se v naši nalogi osredotočili na prakso digitalnega naročanja, ki je v svetovnem pogledu že zelo uspešen proces.

V naši nalogi smo inspiracijo gradili procesa digitalnega naročanja v zdravstvu. Digitalno naročanje v zdravstvu je pokazalo svojo uporabnost in praktičnost v razmahu COVIDA-19. Popularnost je tako narastla, da je Forbes spletno naročanje terminov uvrstil med tri glavne trende v panogi naročanja na termine.

Iz tega smo ustvarili širšo sliko uporabe digitalnega naročanja v praksi in s tem ustvarili teoretično programsko rešitev v praksi, ki bo reševala probleme procesa naročanja stran na termine – kar se lahko pozneje aplicira na različne vrste storitve.

Predstavili smo sledeč problem: Uporabniki storitev, se še vedno v veliki meri na storitve naročamo preko telefonskega klica ali preko elektronske pošte. To velja za skoraj vse storitve od naročanja na zdravniški pregled do enostavnejših storitev kot so na primer frizerske storitve. Pri 'stari praksi' naročanja smo naleteli na ogromno problemov, ki storitev naročanja naredijo manj učinkovito in manj praktično – kar je samo potrdilo koristnost digitalizacije naročanja.

V nadaljevanju smo predstavili konceptualni načrt, skupaj z izdelano podatkovno bazo za in obrazci za potencialno aplikacijo. Na teoretični osnovi smo opisali kako bo aplikacija delovala. Na koncu pa smo ponudili še predloge za izboljšavo aplikacije.

LITERATURA IN VIRI

1. Aimicheva, G., Kopeyev, Z., Ordabayeva, Z., Tokzhigitova, N. & Akimova, S. (2020). A spiral model teaching mobile application development in terms of the continuity principle in school and university education. *Education and Information Technologies*, 25(3), 1875-1889.
2. Akshay, V., Kumar, A., Alagappan, R. M. & Gnanavel, S. (2019). BOOKAZOR-an Online Appointment Booking System. *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)* (str. 1-6). Vellore: IEEE.
3. Ala, A. & Chen, F. (2022). Appointment scheduling problem in complexity systems of the healthcare services: A comprehensive review. *Journal of Healthcare Engineering*, 1-16.
4. Asli, A. (2019, 28. marec). 2019 Online Scheduling And Client Management Trends For Appointment-Based Businesses. *Forbes*. Pridobljeno 3. julija 2022 iz <https://www.forbes.com/sites/forbestechcouncil/2019/03/28/2019-online-scheduling-and-client-management-trends-for-appointment-based-businesses/?sh=1fe3b2622363>
5. Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *International Journal of Engineering & Technology*, 2(5), 1-7.
6. Borderless Technology Corp. (23. november 2017). The Advantages of Digitizing Human Life Experiences. *Medium*. Pridobljeno 2. julija 2022 iz <https://medium.com/@Borderless/the-advantages-of-digitizing-human-life-experiences-aae0e35c7b6e>.
7. Cooper, A., Reimann, R. & Cronin, D. (2014). *About Face: The Essentials of Interaction Design* (4. izd.). Indianapolis: John Wiley & Sons.
8. Croon Fors, A. (2013). The ontology of the subject in digitalization. V R. Luppicini (ur.), *Handbook of research on technoself: Identity in a technological society Volume 1* (str. 45-63). Hershey: IGI Global.
9. Gao, J., Bai, X., Tsai, W. T. & Uehara, T. (2014). Mobile application testing: a tutorial. *Computer*, 47(2), 46-55.

10. Goodwill Comunity Foundation, Inc. (brez datuma). *Computer Basics: Understanding Applications*. Pridobljeno 2. julija 2022 iz <https://edu.gcfglobal.org/en/computerbasics/understanding-applications/1/>
11. Gumus, B., Ertas, A., Tate, D. & Cicek, I. (2008). The transdisciplinary product development lifecycle model. *Journal of Engineering Design*, 19(3), 185-200.
12. Hoehle, H. (2015). Mobile application usability. *MIS quarterly*, 39(2), 435-472.
13. Huynh, M. Q., Ghimire, P. & Truong, D. (2017). Hybrid app approach: could it mark the end of native app domination?. *Issues in Informing Science and Information Technology*, 14, 50-65.
14. Indeed Editorial Team. (2021, 23. julij). *What Is an App? Types of Apps and Examples*. Pridobljeno 3. septembra 2022 iz <https://www.indeed.com/career-advice/career-development/what-is-an-app>
15. Jobe, W. (2013). Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies*, 7(4), 27-32.
16. Kaur, A. & Kaur, K. (2015). Suitability of existing software development life cycle (sdlc) in context of mobile application development life cycle (madlc). *International Journal of Computer Applications*, 116(19), 1-6.
17. Malavolta, I., Ruberto, S. T. & Terragni, V. (2015). End users' perception of hybrid mobile apps in the google play store. *IEEE International Conference on Mobile Services* (str. 25-32). Pridobljeno 2. julija 2022 iz http://www.ivanomalavolta.com/files/papers/MS_2015.pdf
18. Mathur, S. & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications*, 1(12), 29-34.
19. Nayebi, F., Desharnais, J. M. & Abran, A. (2012). The state of the art of mobile application usability evaluation. *25th IEEE Canadian Conference on Electrical and Computer Engineering* (str. 1-4). Montreal: IEEE.
20. Nourbakhsh, M., Zin, R. M., Irizarry, J., Zolfagharian, S. & Gheisari, M. (2012). Mobile application prototype for on-site information management in construction industry. *Engineering, construction and architectural management*, 19(5), 474-494.
21. Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G. & Gall, H. C. (2015). How can i improve my app? classifying user reviews for software maintenance and evolution. *2015 IEEE international conference on software maintenance and evolution* (str. 281-290). Bremen: IEEE.
22. Peruzzini, M., Grandi, F. & Pellicciari, M. (2017). Benchmarking of Tools for User Experience Analysis in Industry 4.0. *Procedia manufacturing*, 11, 806-813.
23. QLess, Inc. (brez datuma). How Appointment Scheduling Software Improves Your Customer Experience. Pridobljeno 2. julija 2022 iz <https://qless.com/how-appointment-scheduling-software-improves-your-customer-experience/>
24. Rosenfeld, L. & Morville, P. (2002). *Information architecture for the world wide web*. Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc.
25. Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), 8-13.

26. Stoica, M., Mircea, M. & Ghilic-Micu, B. (2013). Software development: agile vs. traditional. *Informatica Economica*, 17(4), 64-76.
27. Software. (2020, 31. marec). V *Techopedia*. Pridobljeno 27. aprila 2022 iz <https://www.techopedia.com/definition/4356/software>
28. Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. Sebastopol, Kalifornija, ZDA: O'Reilly Media, Inc.
29. Villaumbrosia, C. G. & Anon, J. (2017). *The Product Book: How to Become a Great Product Manager* (2. izd.). San Francisco: Product School Inc.