

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Robi Tomc

Razvoj in namestitvev aplikacije v oblak

DIPLOMSKO DELO
VISOKOŠOLSKE STROKOVNE ŠTUDIJSKE PROGRAMA PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2013

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 00399/2013

Datum: 05.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ROBI TOMC**

Naslov: **RAZVOJ IN NAMESTITEV APLIKACIJE V OBLAK**
CLOUD APPLICATION DEVELOPMENT AND DEPLOYMENT

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Računalništvo v oblaku je področje, s katerim se vsakodnevno srečujemo ob uporabi svetovnega spleta. Za uporabo storitev v oblaku se odloča vedno več uporabnikov ter podjetij, saj ta možnost omogoča, da so nam podatki vedno dosegljivi in zmanjšati stroške poslovanja. V prvem delu diplomske naloge predstavite področje računalništva v oblaku in poiščite nekaj aktualnih storitev za shranjevanja datotek v oblaku. V okviru drugega dela diplome izdelajte aplikacijo za shranjevanje in deljenje datotek v oblaku. Aplikacija naj poleg funkcionalnosti, ki jih ponujajo drugi ponudniki, nudi še dodatne funkcionalnosti za boljši pregled in upravljanje arhiva. Predstavite razvoj aplikacije in njeno namestitev v oblak.

Mentor:

viš. pred. dr. Aljaž Zrnec

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani **Robi Tomc,**

z vpisno številko **63080293,**

sem avtor diplomskega dela z naslovom:

Razvoj in namestitvev aplikacije v oblak

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno, pod mentorstvom viš. pred. dr. Aljaža Zrneca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.) povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____

Podpis avtorja: _____

Zahvala

Za nasvete in vodenje pri pisanju diplomske naloge se zahvaljujem mentorju, viš. pred. dr. Aljažu Zrnecu. Zahvala je namenjena tudi sošolcem in vsem ostalim, ki so kakorkoli pripomogli pri nastanku te diplomske naloge.

Posebno zahvalo pa namenjam staršem, ki so me v času študija spodbujali in finančno podpirali.

Kazalo vsebine

1	Uvod.....	1
2	Računalništvo v oblaku	3
2.1	Kaj je računalništvo v oblaku?.....	3
2.2	Lastnosti računalništva v oblaku.....	4
2.2.1	Prednosti.....	4
2.2.2	Slabosti	5
2.3	Tipi računalništva v oblaku.....	6
2.3.1	Javni oblak.....	6
2.3.2	Zasebni oblak	6
2.3.3	Oblak skupnosti.....	6
2.3.4	Hibridni oblak	7
2.4	Predstavitev storitvenih modelov.....	8
2.4.1	Infrastruktura kot storitev	8
2.4.2	Platforma kot storitev	8
2.4.3	Programska oprema kot storitev.....	9
3	Shranjevanje datotek v oblaku	11
3.1	Opis funkcionalnosti	11
3.2	Predstavitev ponudnikov za shranjevanje datotek	11
3.3	Motivacija za razvoj lastne aplikacije.....	14
4	Razvoj spletne aplikacije.....	17
4.1	Namen aplikacije	17
4.2	Predstavitev aplikacije	19
4.3	Zgradba aplikacije.....	21
4.4	Predstavitev uporabljenih tehnologij	21
4.5	Podatkovna baza	25
4.6	Struktura map za shranjevanje datotek	27
4.7	Izzivi in zapleti med razvojem.....	28
4.8	Možne izboljšave	31
5	Namestitev spletne aplikacije.....	35
5.1	Platforma kot storitev.....	35
5.2	Predstavitev ponudnikov platform.....	35
5.3	Namestitev aplikacije.....	37
5.3.1	Pridobitev SQL in SMTP strežnika.....	38
5.3.2	Namestitev na AppHarbor.....	39

5.3.3	Namestitev na Apprendo	40
5.3.4	Namestitev na Uhuru AppCloud	42
5.4	Namestitev podatkovne baze.....	43
5.5	Nastavitev spletne aplikacije	44
5.6	Vzdrževanje spletne aplikacije.....	47
6	Zaključek	49
7	Priloge.....	51
8	Viri in literatura	61

Kazalo slik

Slika 1: Arhitektura oblaka.....	3
Slika 2: Tipi računalništva v oblaku.....	7
Slika 3: Sklad storitvenih modelov.....	9
Slika 4: Diagram primerov uporabe aplikacije.....	18
Slika 5: Prijavna stran spletne aplikacije.....	19
Slika 6: Uporabniški vmesnik z brskalnikom datotek.....	20
Slika 7: Zgradba spletne aplikacije.....	21
Slika 8: Konceptualni model podatkovne baze.....	26
Slika 9: Drevesna struktura map uporabnikov.....	27
Slika 10: Primer opozorila in pripadajoče programske kode.....	31
Slika 11: Prilagojen uporabniški vmesnik za izbiro datotek.....	32
Slika 12: Predloga ASP.NET aplikacije za mobilne naprave.....	34
Slika 13: Namestitev aplikacije v oblak.....	37
Slika 14: Ustvarjanje aplikacije na platformi AppHarbor.....	38
Slika 15: Aplikacija nameščena na AppHarbor.....	39
Slika 16: Struktura map in datotek projekta.....	40
Slika 17: Nameščanje aplikacije na platformo Apprenda.....	41
Slika 18: Uporabniški vmesnik Uhuru AppCloud.....	42
Slika 19: Izdelava SQL skripte z orodjem Visual Studio.....	43
Slika 20: Vzpostavitev povezave z oddaljenim SQL strežnikom.....	44
Slika 21: Začetna stran s prikazom informacij o rabi storitve.....	45
Slika 22: Splošne nastavitve aplikacije.....	46
Slika 23: Sekvenčni diagram za primer kopiranja mape.....	52
Slika 24: Sekvenčni diagram za primer registracije uporabnika.....	53
Slika 25: Sekvenčni diagram za primer spreminjanja nastavitve.....	54
Slika 26: Sekvenčni diagram za primer preimenovanja datoteke.....	55
Slika 27: Sekvenčni diagram za primer nalaganja datoteke.....	56
Slika 28: Sekvenčni diagram za primer prenosa datoteke s spleta.....	57
Slika 29: Sekvenčni diagram za primer deljenja datotek z drugimi uporabniki.....	58
Slika 30: Sekvenčni diagram za primer urejanja vsebine datoteke.....	59

Razlaga uporabljenih kratic

.NET – Programska komponenta sistema Microsoft Windows, ki vsebuje orodja in knjižnice za razvoj programske opreme.

ASCII – American Standard Code for Information Interchange. Standardni nabor znakov (kodna tabela), ki se uporablja za prenos informacij.

ASP – Active Server Pages. Programski jezik, ki se uporablja za izdelavo spletnih strani.

CORS – Cross-Origin Resource Sharing. Mehanizem, ki spletni strani omogoča deljenje virov z drugo domeno.

CSS – Cascading Style Sheets. Predloge v obliki opisnega jezika, ki se uporabljajo za določanje lastnosti (postavitev, oblikovanje) HTML elementov.

GUI – Graphical User Interface. Grafični uporabniški vmesnik, ki skrbi za komunikacijo med uporabnikom in računalnikom.

HTML – HyperText Markup Language. Hipertekstovni označevalni jezik, ki se uporablja za izdelavo spletnih strani.

IaaS – Infrastructure as a Service. Infrastruktura v obliki storitve.

IDE – Integrated Development Environment. Programska oprema, ki omogoča urejanje, prevajanje in razhroščevanje programske kode.

IP – Internet Protocol. Protokol, ki skrbi za prenos podatkov v računalniškem omrežju.

IP naslov – 32-bitno število, ki enolično označuje računalnik v omrežju. Običajno je zapisan s štirimi osembitnimi števili v desetiškem številskem sistemu.

JS – JavaScript. Enostaven skriptni jezik, ki spletni strani omogoči interaktivnost.

MD5 – Message-Digest algorithm 5. Kodirni algoritem (zgoščevalna funkcija), ki se uporablja za shranjevanje gesel in preverjanje datotek.

MIME – Multipurpose Internet Mail Extension. Standard za opisovanje neznakovnih podatkov. Pogosto se uporablja za pošiljanje datotek preko elektronske pošte.

PaaS – Platform as a Service. Platforma v obliki storitve.

SMTP – Simple Mail Transfer Protocol. Enostaven protokol, ki se uporablja za prenos elektronske pošte.

SaaS – Software as a Service. Programska oprema v obliki storitve.

SQL – Structured Query Language. Strukturirani povpraševalni jezik, ki se uporablja za delo s podatkovnimi bazami.

URL naslov – Uniform Resource Locator. Enolični naslov spletne strani v svetovnem spletu.

Povzetek

Računalništvo v oblaku je področje, s katerim se vsakodnevno srečujemo ob uporabi svetovnega spleta. Za uporabo storitev v oblaku se odloča vedno več uporabnikov in podjetij, saj ta možnost omogoča, da so podatki vedno dosegljivi, in pomaga zmanjšati stroške poslovanja.

V prvem delu diplomske naloge smo na splošno predstavili področje računalništva v oblaku in opisali nekaj aktualnih storitev za shranjevanje datotek v oblaku. V nadaljevanju smo predstavili razvoj lastne aplikacije za shranjevanje in deljenje datotek, pri čemer smo se poskušali približati že obstoječim rešitvam. Omenjena aplikacija poleg standardnih ponuja še nekaj dodatnih funkcionalnosti za boljši pregled in upravljanje arhiva. Raziskali smo tudi možnosti za namestitev aplikacije na ustrezno platformo. Predstavili smo nekaj ponudnikov in opisali postopek same namestitve v oblak.

Ključne besede:

Računalništvo v oblaku, spletna aplikacija, svetovni splet

Abstract

Cloud computing is a topic we encounter in everyday use of the World Wide Web. More and more users and companies decide to use the services in the cloud because this option ensures that our data is always available to us and helps reduce operating costs.

In the first part of the thesis we provide a general explanation of cloud computing and describe some actual services for storing files in the cloud. We also present the development of our own application for storing and sharing files, which we tried to approach to existing solutions. In addition to standard functionalities, our application offers functionalities for better overview and management of the archive. We explore the possibilities for deploying the application on the appropriate platform. We present some service providers and describe the process of deploying it in the cloud.

Keywords:

Cloud computing, web application, World Wide Web

1 Uvod

Tekom razvoja računalniške tehnologije smo se srečevali z različnimi mediji za shranjevanje in prenos podatkov. Namesto kaset danes uporabljamo zgoščenke. Za predvajanje glasbe uporabljamo računalnik in medmrežje. Običajno pošto izpodriva elektronska pošta. Namesto časopisov vse bolj prebiramo novice na spletu. Vse te spremembe so nastale z namenom, da bi pridobili hitrost in večjo kapaciteto za shranjevanje podatkov.

Marsikdo misli, da je računalništvo v oblaku nekaj novega, vendar začetki le-tega segajo v leto 1969 [3], ko je nastala ideja, da bi bili vsi podatki shranjeni na enem mestu. Prav iz te zamisli je nastal ARPANet (akronim za Advanced Research Projects Agency Network), prvo medmrežje, katerega glavni namen je bil dostop do podatkov iz oddaljenih računalnikov. Kot že ime pove, je bilo prvotno namenjeno agencijam za napredne projekte. Iz takratnega ARPANet se je razvil Internet, splet pa je nastal na podlagi predloga Tima Bernersa Leeja.

Od nastanka prvega medmrežja se je marsikaj spremenilo. Eden prvih mejnikov v zgodovini računalništva v oblaku je CERN-ova objava prve spletne strani, 6. avgusta 1991 [3]. Le nekaj let pozneje (leta 1999) se je pojavil Salesforce.com – prva spletna stran s konceptom zagotavljanja poslovnih aplikacij. Leta 2002 je podjetje Amazon predstavilo svoj nabor storitev, ki vključuje skladiščenje podatkov uporabnikov in obračunavanje porabe sredstev. S tem so nastali temelji računalništva v oblaku, v taki obliki, kot ga poznamo danes.

Področje računalništva v oblaku se nenehno razvija in je med uporabniki in tudi med podjetji vse bolj priljubljeno, zaradi česar smo se odločili, da ga v okviru diplomske naloge podrobneje predstavimo. Osnovni namen diplomskega dela je predstavitev razvoja in namestitve spletne aplikacije v oblak. Zaradi boljšega razumevanja je uvodno poglavje namenjeno predvsem teoretičnim osnovam računalništva v oblaku. Preostali del diplomske naloge opisuje uporabo, razvoj in postavitve lastne aplikacije za shranjevanje datotek v oblak.

Kot smo že omenili, se s področjem računalništva v oblaku vsakodnevno srečujemo. Dokaz za to so številne storitve na svetovnem spletu. Poiskali smo nekaj aktualnih ponudnikov storitev in jih glede na njihov namen razvrstili po kategorijah. Oglejmo si nekaj primerov:

- a) spletno shranjevanje in arhiviranje podatkov: SkyDrive, iCloud, Amazon Cloud Drive, Dropbox, Box, Mozy, Carbonite;
- b) storitve za hitro deljenje datotek: CloudApp, Dropir, Skitch, SendSpace;
- c) elektronska pošta in pisarniška opravila: Google Documents, Google Mail, MSN Live, Yahoo;

- d) spletna družbena omrežja: Google+, Facebook, Twitter, Netlog, MySpace, LinkedIn;
- e) delo z zvokom in videom: Google Play, YouTube, iTunes Store, Spotify, Pandora, Grooveshark, SoundCloud, Deezer, DailyMotion, MetaCafe, Vimeo;
- f) delo s slikami in spletni albumi: Flickr, Google Picasa, OpenPhoto.

Že od pojava svetovnega spleta se je napovedovalo, da je prihodnost računalništva na spletu. Računalništvo v oblaku je to napoved uresničilo: vse, kar je bilo nekoč mogoče početi le z uporabo namiznih aplikacij, lahko danes počnemo na spletu.

Nekaj drugih primerov računalništva v oblaku: programiranje na spletu in združevanje programske kode, spletni atlasi in zemljevidi, spletne enciklopedije, prevajalniki in slovarji, storitve za pretvorbo datotek med različnimi formati (iz .docx v .doc, iz .mp4 v mp3 itd).

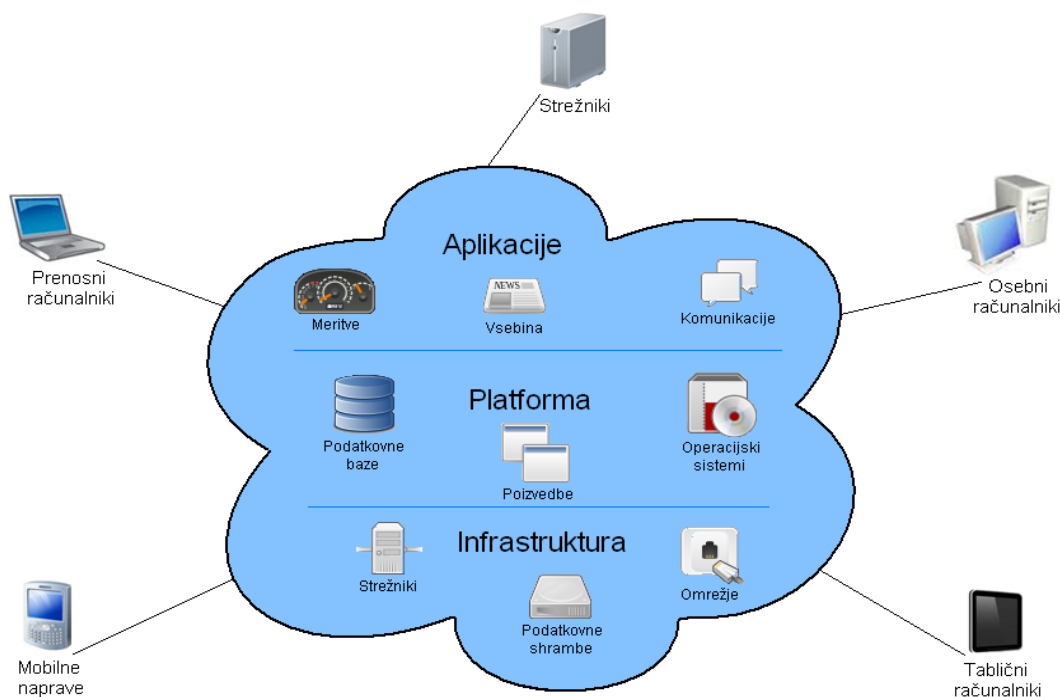
2 Računalništvo v oblaku

2.1 Kaj je računalništvo v oblaku?

Računalništvo v oblaku je slog računalništva, ki proučuje rabo računalniških sredstev (strojne in programske opreme) v obliki storitev v svetovnem spletu. Razlog, zakaj področje imenujemo ravno računalništvo v oblaku (angl. cloud computing), se skriva v delovanju in organizaciji računalniških omrežij. Vsako delujoče omrežje lahko povežemo v neko drugo omrežje in tako dobimo novo, večjo celoto. Za ponazoritev sheme omrežja so se morali inženirji domisliti načina, ki bi predstavljal povezanost omrežij. Po drugi strani pa so omrežje hoteli predstaviti kot nekaj velikega, saj so ga težko razumeli. Takrat je nastala ideja o oblaku [4] kot metafora za takratno telefonsko omrežje.

Oblaki podpirajo različne storitvene modele (Slika 1). To so infrastruktura, platforma in programska oprema. Vsak del je enako pomemben, saj s svojim delovanjem podpira naslednji nivo in tako prispeva k funkcionalnosti oblaka. Vsakega od storitvenih nivojev bomo natančneje predstavili v nadaljevanju.

Glavni namen računalništva v oblaku je dostop do storitev, pri čemer uporabniku ni treba vedeti, kje se strojna oprema, ki uresničuje njegove zahteve, nahaja. Celoten oblak je lahko realiziran na različne načine, zato ločimo več tipov računalništva v oblaku.



Slika 1: Arhitektura oblaka.

Oblak je alternativa nakupu aplikacij, storitev in infrastrukture [5]. Podjetjem in uporabnikom storitev ni treba poganjati na lokalnem računalniku, saj se lahko odločijo za najem strežnika in tako izkoristijo prednosti oblaka. Najem strežnika oziroma storitev je izveden s pomočjo virtualizacije, zato je možno izkoriščati fleksibilnost – uporabnik najame tolikšno zmogljivost, kot jo potrebuje in jo kasneje po potrebi poveča. To je prednost tudi za manjša in novonastala podjetja, saj običajno še nimajo infrastrukture, postavitev lastne pa bi lahko predstavljala prevelik strošek.

Ko govorimo o računalništvu v oblaku, se srečujemo z dvema vlogama. Računalniku, ki daje na voljo svojo strojno in programsko opremo, pravimo strežnik. Naprave in programsko opremo, s pomočjo katerih dostopamo do sredstev, pa imenujemo odjemalci.

2.2 Lastnosti računalništva v oblaku

Čeprav računalništvo v oblaku prinaša številne prednosti, ima tudi določene slabosti. Ker smo v sklopu diplomske naloge razvili aplikacijo za shranjevanje datotek, smo poiskali lastnosti z vidika podatkov (povzeto po [6,7]).

2.2.1 Prednosti

Možna razširitev zmogljivosti virov. Strojna oprema sčasoma postane tehnološko zastarela (počasnost, dolgi odzivni časi, premajhna kapaciteta pomnilnika). Pri uporabi storitev v oblaku lahko zmogljivost virov kadarkoli nadgradimo glede na naše potrebe. Stroški razširitve so nižji od nakupa nove strojne opreme.

Sinhronizacija med napravami. Računalništvo v oblaku je mogoče uporabljati kot sredstvo za sinhronizacijo. S pomočjo programske opreme imamo lahko dostop do podatkov s kateregakoli računalnika ali mobilne naprave. Vsebina map in datotek je enaka na vseh napravah, ki jih sinhroniziramo.

Zagotovljen enostaven, univerzalen dostop. Do podatkov v oblaku lahko dostopamo od koderkoli in kadarkoli. Za prenos datotek ne potrebujemo prenosljivih medijev, dovolj je naprava z dostopom do svetovnega spleta. Za uporabo storitve običajno ni treba nameščati dodatne programske opreme, saj potrebujemo le spletni brskalnik.

Varnost podatkov glede na okvare. V primerjavi s hranjenjem podatkov na lokalnem računalniku so naši podatki v oblaku bolj varni, saj je verjetnost odpovedi strojne opreme

oblaka manjša v primerjavi z lokalno strojno opremo. Prenosni mediji so občutljivi na okolščine (izpostavljenost vlagi, visokim temperaturam, magnetnemu polju itd.), pri strojni opremi pa lahko pride do okvare.

Možnost deljenja in skupne rabe. Datoteke lahko prenaša ali spreminja več uporabnikov hkrati. Na ta način lahko oblak izkoristimo kot prenosni medij. Skupno rabo dokumentov lahko zelo dobro izkoristimo v primeru dela v ekipi.

Pregledovanje po verzijah. Vsaka sprememba datoteke se zabeleži, zato lahko pregledamo prejšnje verzije ali spremembe kadarkoli razveljavimo. Ni nam treba ustvarjati dodatnih datotek, da bi imeli nadzor nad spremembami.

Manjši stroški za podjetja. Manjšim in novonastalim podjetjem lahko računalništvo v oblaku predstavlja alternativo nakupu storitev in infrastrukture, kar je cenejša rešitev kot postavitve lastne. Ko se podjetje bolj razvije, lahko nadgradi obstoječo zmogljivost.

2.2.2 Slabosti

Nepooblaščen dostop oz. možnost napada. Do podatkov, shranjenih na svetovnem spletu, ima dostop vsak, ki pozna geslo. V primeru kraje prijavnih podatkov so podatki ogroženi. Ker se podatki prenašajo preko spleta, so možni številni napadi (prestrezanje in ponarejanje sporočil, kraja piškotkov itd.).

Zaupanje ponudniku storitve. Preden se odločimo za uporabo oblaka, moramo zaupati ponudniku storitve. Ker ponudnika storitve načeloma ne poznamo, odločitev za uporabo oblaka predstavlja tveganje. Pri hranjenju in obdelavi podatkov so izpostavljeni predvsem naši osebni in finančni podatki.

Odvisnost od omrežja in ponudnika dostopa do spleta. Dosegljivost naših datotek lahko marsikaj prepreči. Če nismo povezani v medmrežje, potem do podatkov ne moremo. Tudi v primeru izpada strežnika oziroma nedelovanja storitve do podatkov ne moremo.

Podprti tipi datotek. Nekatere storitve podpirajo samo določene tipe datotek. Če storitev podpira samo dokumente (besedilo, preglednice, predstavitve), potem ne moremo shranjevati glasbe, videa, programov in drugih tipov datotek.

Odzivnost in hitrost prenosa podatkov. Ker so naše datoteke shranjene na internetu, je potreben daljši dostopni čas. Pri velikih datotekah lahko traja zelo dolgo, da jih prenesemo na svoj računalnik. Hitrost prenosa je zato počasnejša v primerjavi z lokalno strojno opremo.

2.3 Tipi računalništva v oblaku

Glede na potrebe se podjetja in uporabniki lahko odločajo med štirimi različnimi tipi računalništva v oblaku (Slika 2). Osnovna razlika med posameznimi tipi je način izvedbe oblaka. Vsak tip je podrobneje predstavljen v nadaljevanju (povzeto po [1,2]).

2.3.1 Javni oblak

Javni oblak (angl. public cloud) temelji na standardnem modelu računalništva v oblaku, kjer ponudnik storitve nudi svoja sredstva preko svetovnega spleta. Kapaciteta za shranjevanje podatkov in pasovna širina sta neomejeni, uporabnik pa ju lahko izkoristi le do te mere, kolikor ima zakupljeno. Ponudnik lahko uporabo virov obračunava po principu »pay-per-use« (plačaj, kar uporabljaš; pogosto se uporablja tudi izraz »pay-as-you-go«) ali pa se odloči za brezplačno ponudbo. Javni oblak omogoča strankam uporabo programske opreme ali prostor za namestitve lastnih aplikacij. Primeri javnih oblakov so naslednji ponudniki storitev: Amazon EC2, Google App Engine, Windows Azure, IBM Blue Cloud.

2.3.2 Zasebni oblak

Zasebni oblak (angl. private cloud) ali tudi notranji oblak temelji na podlagi virtualizacije. Primeren je za podjetja, ki želijo imeti svoj oblak, saj je ponudnik storitve na voljo le enemu podjetju, ki plačuje uporabo virov. V primerjavi z javnim oblakom je zasebni oblak varnejši, vendar prinaša tudi višje stroške. Glede na način upravljanja oblaka, ločimo dve vrsti zasebnih oblakov. Uporaba zasebnega oblaka je namenjena predvsem podjetjem in organizacijam, saj je infrastruktura lahko locirana v sami zgradbi. V tem primeru je upravljanje oblaka lahko prepuščeno organizaciji sami. Ni pa to nujno, saj upravljanje lahko prepustimo tretji osebi. Primeri zasebnega oblaka: VMware vCloud, Citrix VDI, Novell Cloud Manager.

2.3.3 Oblak skupnosti

Oblak skupnosti (angl. community cloud) oziroma vertikalni oblak je namenjen organizacijam iz določene skupnosti, ki imajo skupni cilj. Vse organizacije, ki si delijo infrastrukturo, imajo isto skrb za varnost, skladnost in pristojnost. Ker je infrastruktura deljena med več organizacij, se tudi cena razdeli, zato so za posamezno organizacijo stroški nižji. Oblak je lahko lokalni ali pa se nahaja nekje drugje. O lokalnem oblaku govorimo, kadar je za

upravljanje zadolžena ena od organizacij. Lahko pa se odločimo za alternativno rešitev in skrb prepustimo tretji osebi.

2.3.4 Hibridni oblak

Hibridni oblak (angl. hybrid cloud) je kombinacija dveh ali več oblakov različnih tipov. Združimo lahko zasebni in javni oblak ter uporabljamo prednosti obeh. Vsi posamezni oblaki so med seboj povezani tako, da podatke in programe uporabljamo v vseh modelih. Prednost uporabe hibridnega oblaka se izkaže v možnosti razporejanja aktivnosti. Če podjetje uporablja hibridni oblak, se lahko odloči, katere aktivnosti bodo uporabljene v zasebnem oblaku in katere v javnem. Arhitektura hibridnega oblaka zahteva postavitev lokalnih virov in oddaljene strežniške infrastrukture.



Slika 2: Tipi računalništva v oblaku.

(povzeto po: <http://www.armedia.com/blog/2012/03/federal-cloud-computing-challenges-part-1-cloud-deployment-models/>)

2.4 Predstavitev storitvenih modelov

Kot smo že omenili, oblak podpira različne storitvene modele, ki se med seboj razlikujejo glede na to, kaj omogočajo in komu so namenjeni. Na takšen način je mogoče izvesti virtualizacijo in upravljanje s sredstvi strojne in programske opreme.

2.4.1 Infrastruktura kot storitev

Infrastruktura kot storitev (angl. Infrastructure as a Service – IaaS) zagotavlja vire mrežne in strojne opreme in tako podpira ostale plasti oblaka. Gre za najnižji nivo v organizaciji storitvenih modelov. IaaS je namenjena vsem uporabnikom in podjetjem, ki želijo najeti strežnik oziroma druge vire strojne opreme [9].

Infrastruktura je izvedena v obliki virtualizacije, kar pomeni, da naročnik dobi le omejen del vira, ki je lahko procesorska moč, pomnilnik ali diskovni prostor. Poraba virov nekega uporabnika ne vpliva na druge uporabnike. Operacijski sistem in strojno opremo uporabljamo, kot bi jo imeli pri sebi, le da imamo zakupljeno določeno zmogljivost. Takšen pristop omogoča razširljivost oblaka, saj mora ponudnik le popraviti omejitve virtualizacije. Druga prednost se izkaže v obračunavanju virov, saj plačujemo le obremenitev procesorja, porabljen prostor itd. Najem infrastrukture predstavlja prednost predvsem za manjša podjetja, saj je taka možnost cenejša v primerjavi z nakupom celotne strojne opreme, zaradi česar je začetna investicija manjša.

Med najbolj znane ponudnike IaaS uvrščamo Amazon AWS, Savvis Symphony, Teramarks VCloud Express [2].

2.4.2 Platforma kot storitev

Platforma kot storitev (angl. Platform as a Service – PaaS) je vmesni člen oblaka in programski opremi omogoča rabo strojne opreme. Nivo PaaS je namenjen razvijalcem programske opreme, saj jim nudi dostop do infrastrukture in omogoča postavitev lastne programske opreme, ki bo na voljo končnim uporabnikom. Razvijalci programja so pri razvoju omejeni na določen programski jezik, operacijski sistem, podatkovno bazo in vrsto spletnega strežnika [8]. Poleg navedenih lahko ponudnik postavi še druge omejitve.

PaaS je običajno realizirana s pomočjo virtualizacije infrastrukture. Z njenim najemom uporabnik dobi nadzor nad zakupljenimi sredstvi, medtem ko za vzdrževanje še vedno skrbi

ponudnik storitve. Za razvijalce programske opreme PaaS predstavlja fleksibilnost, saj pri razvoju niso omejeni na svoj operacijski sistem, pač pa na okolje, ki jim ga nudi ponudnik.

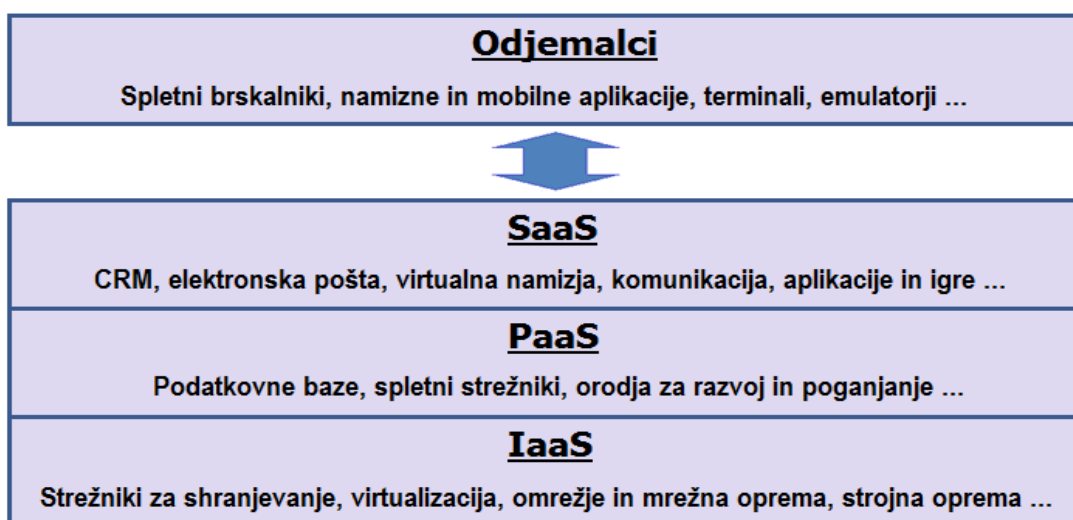
Med najbolj znane ponudnike PaaS vrščamo CloudFoundry, Windows Azure Cloud, Google App Engine, Force.com [2].

2.4.3 Programska oprema kot storitev

Programska oprema kot storitev (angl. Software as a Service – SaaS) uporabnikom nudi dostop do programske opreme (storitev) v oblaku. Ta del je uporabnikom najbolj poznan, saj se z njim vsakodnevno srečujejo. Primere programske opreme predstavljajo npr. Gmail, YouTube, Windows Live Hotmail, Facebook, Twitter itd.

Nadzor uporabnika nad aplikacijami je zelo omejen. V nasprotju s programsko opremo na lokalnih računalnikih, uporabnikom te ni potrebno nameščati, vzdrževati, nadgrajevati ali skrbeti za licenco. Prav tako ni možen razvoj lastne programske opreme, saj je ta že razvita. Uporabnik mora samo pri ponudniku izbrati aplikacije (storitve), do katerih lahko dostopa bodisi preko spletnega uporabniškega vmesnika bodisi preko ustreznega odjemalca. Programska oprema je lahko brezplačna, lahko pa se uporabnikom zaračunava glede na uporabo (angl. pay-per-use) [8].

Za predstavitev oblakov se uporablja 3-nivojska arhitektura (sklad), ki vsebuje vse opisane storitve (Slika 3). V primerjavi z lokalnim sistemom IaaS predstavlja operacijski sistem, SaaS pa programsko opremo. Uporabniki za uporabo aplikacij potrebujejo ustreznega odjemalca.



Slika 3: Sklad storitvenih modelov.

(povzeto po: https://upload.wikimedia.org/wikipedia/commons/3/3c/Cloud_computing_layers.png)

3 Shranjevanje datotek v oblaku

3.1 Opis funkcionalnosti

Poleg shranjevanja datotek ponudniki storitev omogočajo še dodatne funkcionalnosti. V nadaljevanju je predstavljenih nekaj funkcionalnosti, ki smo jih upoštevali tudi pri razvoju naše spletne aplikacije.

Osnovne funkcije nad shranjenim arhivom. Datoteke v arhivu lahko spreminjamo kot na lokalnem računalniku. Lahko ustvarjamo mape, datoteke lahko premikamo in kopiramo, brišemo in obnovljamo, preimenujemo itd.

Pregledovanje in urejanje shranjenih datotek. V oblaku shranjene datoteke lahko pregledujemo in urejamo, ne da bi jih morali prej prenesti na računalnik. Za ogled datotek ne potrebujemo programske opreme na računalniku. Razlika med ponudniki storitev je le v podpori datotek – nekateri so usmerjeni na pisarniške dokumente (besedilo, preglednice, predstavitve), drugi imajo podporo za multimedijske datoteke itd.

Nadzor nad različnimi verzijami datoteke. Spremljamo lahko spremembe vsebine datotek. Če isto datoteko ureja več uporabnikov, vemo, kdo, kdaj in kaj je spremenil. Vsako spremembo lahko kadarkoli razveljavimo.

Deljenje datotek z drugimi uporabniki. Uporabniki lahko svoje datoteke delijo z ostalimi uporabniki. Deljenje datotek lahko razdelimo na dva sklopa:

- a) Souporaba z uporabniki storitve: Datoteko lahko spreminjajo le določeni uporabniki storitve. Lastnik datoteke določi uporabnike, ki lahko dostopajo do datoteke.
- b) Deljenje s komerkoli v spletu: Datoteko lahko prenese na računalnik kdorkoli, ki ima dostop do spleta (ni nujno, da je registriran uporabnik storitve). Datoteka je dosegljiva preko posebnega URL naslova.

3.2 Predstavitev ponudnikov za shranjevanje datotek

Storitev za shranjevanje datotek v oblaku je ogromno, zato smo poiskali le nekaj aktualnih in preučili njihove lastnosti. Opis ponudnikov storitev za shranjevanje sledi v nadaljevanju.

Dropbox

Dropbox je trenutno ena najbolj znanih storitev za shranjevanje datotek. Nastal je leta 2008 in ima danes čez 100 milijonov uporabnikov. Preko spletnega uporabniškega vmesnika, ki je na voljo v 16 jezikih, lahko shranjujemo do 300 MB velike datoteke [10]. Za prenos večjih datotek in sinhronizacijo med napravami je na voljo programska oprema, ki deluje na večini operacijskih sistemov in mobilnih naprav (Microsoft Windows, Mac OSX, Linux, Android, BlackBerry, Symbian, iOS). Jedro spletne aplikacije in programska oprema sta napisana v programskem jeziku Python. Za zaščito uporabniških podatkov se uporablja 256-bitno SSL in AES kodiranje.

Dropbox ponuja 3 vrste uporabniških računov [11]. Vsak uporabnik se lahko registrira z brezplačnim uporabniškim računom, s čimer dobi 2 GB prostora. Z vsakim povabilom novega uporabnika obstoječi uporabnik pridobi dodatnih 500 MB. Brezplačni račun je možno kadarkoli nadgraditi na t. i. Pro različico. Tukaj uporabniki izbirajo med kapacitetami 100, 200 in 500 GB (te kapacitete zaračunavajo). Za podjetja je na voljo Team različica, ki ponuja nad 1 TB prostora za do 5 uporabnikov, vključno z orodji za administratorja.

Box

Box je spletna storitev za arhiviranje, sinhronizacijo in deljenje datotek. Storitvi se je od začetka delovanja leta 2005 pridružilo več kot 150.000 podjetij in skupno 15 milijonov uporabnikov [12]. Programska oprema za sinhronizacijo podpira operacijski sistem Microsoft Windows ter mobilne naprave Android, BlackBerry, iOS ter WebOS. Spletni uporabniški vmesnik omogoča tudi pregledovanje in urejanje datotek, z brezplačnim računom pa lahko nalagamo do 250 MB velike datoteke. Celotna spletna storitev je napisana v programskih jezikih PHP, Java ter Scala.

Ponudnik Box ponuja 3 vrste uporabniških računov [13]. Za osebno rabo je na voljo brezplačni račun s kapaciteto 5 GB, ki ga lahko nadgradimo do 50 GB. Z nadgraditvijo dobimo tudi nekatere druge funkcionalnosti. Podjetja lahko izbirajo med računoma Business in Enterprise. Prvi prinaša kapaciteto 1 TB in večino funkcij, drugi pa ima neomejeno kapaciteto in podprte vse funkcije.

Google Drive

Google Drive je Googlova storitev za shranjevanje in sinhronizacijo dokumentov, ki je začela delovati aprila 2012. Uporabniški vmesnik je na voljo v več jezikih in podpira več kot 30 vrst

datotek, ki jih je z uporabo storitve Google Dokumenti mogoče pregledovati in urejati kar na spletu. Spremembe datotek se beležijo, pregledujemo lahko spremembe zadnjih 30 dni. Aplikacija za sinhronizacijo dokumentov podpira operacijske sisteme Microsoft Windows, Mac OSX, Google Chrome OS in Android.

Za uporabo storitve moramo imeti Google uporabniški račun, s katerim uporabljamo tudi druge izdelke ponudnika Google. Vsak uporabnik ima na voljo 5 GB prostora, ki ga lahko uporabi za shranjevanje dokumentov ali slik v storitvi Google Picassa ali Google+ Slike. Shranjene dokumente lahko uporabljamo tudi v drugih Googlovih izdelkih. Po naročilu lahko uporabnik obstoječo kapaciteto poveča [14].

Microsoft SkyDrive

Microsoft SkyDrive (prej imenovana Windows Live Folders) je storitev podjetja Microsoft. Nastala je avgusta 2007 in v tem času pridobila več kot 250 milijonov uporabnikov [15]. Dokumente, ki niso večji od 300 MB, lahko shranjujemo tudi preko spletnega uporabniškega vmesnika. Za prenos večjih dokumentov oziroma za sinhronizacijo pa moramo namestiti programsko opremo. Namizna aplikacija podpira operacijske sisteme Microsoft Windows in Mac OSX ter nekatere mobilne naprave. Spletni vmesnik temelji na tehnologiji HTML5, uporabniški vmesnik pa je na voljo v 94 jezikih [16].

Z brezplačno registracijo uporabnik pridobi 7 GB prostora za shranjevanje, s plačilom pa se lahko kapaciteta razširi do 25 GB. Dodatki Office Web Apps uporabnikom omogočajo ustvarjanje, pregledovanje in urejanje dokumentov paketa Microsoft Office kar na spletu. Uporabniki Dropboxa lahko svoje datoteke upravljajo tudi preko storitve SkyDrive.

Ubuntu One

Ubuntu One je spletna storitev podjetja Canonical Ltd. Razvita je bila maja 2009 (prva verzija je izšla z operacijskim sistemom Ubuntu 9.04) kot sredstvo za sinhronizacijo med napravami z operacijskim sistemom Ubuntu Linux. Shranjene datoteke lahko delimo tudi z drugimi uporabniki storitve. Programska oprema za sinhronizacijo je na voljo tudi za operacijske sisteme Microsoft Windows, Mac OSX, iOS in Android. Spletni uporabniški vmesnik je napisan v programskem jeziku Python [17].

Uporabnikom je na voljo brezplačni uporabniški račun s kapaciteto 5 GB, ki pa jo lahko povečujemo po 500 MB z vsakim povabilom novega uporabnika. Uporabniški račun je

možno kadarkoli nadgraditi, s čimer si lahko uporabniki sami izberejo kapaciteto. Poleg datotek lahko uporabniki shranjujejo tudi e-kontakte, z dodatkom Photos pa lahko svoje shranjene slike predstavijo v obliki spletnih albumov. Dodatek Music Store omogoča uporabnikom nakupovanje elektronske glasbe. Z nakupom glasbe uporabnik pridobi dodatnih 20 GB prostora [18].

Apple iCloud

iCloud je Applova spletna storitev za shranjevanje datotek v oblaku. Predstavlja alternativo Microsoftovega SkyDrivea, vendar je namenjena uporabnikom operacijskega sistema Mac OSX (podpira pa tudi Microsoft Windows). Storitve je začela obratovati junija 2011, v aprilu 2013 pa so zabeležili 300 milijonov uporabnikov.

Kapaciteta 5 GB omogoča shranjevanje datotek, sinhronizacijo z Applovimi napravami in deljenje z drugimi uporabniki. Z uporabo dodatka iTunes lahko uporabniki predvajajo svojo glasbeno in video zbirko na katerikoli Apple napravi. V storitev so integrirane tudi nekatere aplikacije, kot so elektronska pošta in koledarji [19].

Amazon Cloud Drive

Podjetje Amazon je marca 2011 predstavilo Amazon Cloud Drive, svojo storitev za shranjevanje, deljenje in sinhronizacijo datotek. Programska oprema za sinhronizacijo deluje na operacijskih sistemih Microsoft Windows in Mac OSX. Poleg storitve Cloud Drive je uporabnikom na voljo še Cloud Player, ki je aplikacija, ki omogoča predvajanje in urejanje shranjenih glasbenih datotek. Tako imajo uporabniki dostop do svoje glasbe iz kateregakoli računalnika ali Android naprave – potrebujejo le dostop do spleta [20].

Brezplačno lahko vsak uporabnik pridobi uporabniški račun s kapaciteto 5 GB, ki jih je mogoče nadgraditi do 1 TB. Cena za vsak dodaten gigabajt za eno leto znaša 0,5 USD.

3.3 Motivacija za razvoj lastne aplikacije

Čeprav je storitev za shranjevanje datotek zelo veliko, se njihove funkcionalnosti prav tako zelo razlikujejo. Kot pomanjkljivosti lahko izpostavimo podporo programske opreme za različne operacijske sisteme – nekateri operacijski sistemi namreč niso podprti, zato

samodejna sinhronizacija ni možna. Še vedno pa lahko take storitve uporabljamo preko spletnega vmesnika. Prednost pa je način širjenja storitve med uporabniki – s povabilom novih uporabnikov razširimo kapaciteto, hkrati pa ponudnik pridobi nove uporabnike.

Shranjevanje datotek je le ena od storitev, ki jih ponuja računalništvo v oblaku. Splet je vse bolj dostopen, zato lahko storitve za shranjevanje datotek uporabljamo namesto prenosljivih medijev, če le imamo dostop do spleta. Za razvoj lastne aplikacije smo se odločili z namenom, da uporabnikom predstavimo delovanje teh storitev.

4 Razvoj spletne aplikacije

4.1 Namen aplikacije

V uvodnem poglavju smo našli nekaj ponudnikov za shranjevanje datotek in navedli osnovne funkcionalnosti. Razvili smo tudi lastno programsko opremo kot storitev (SaaS) za shranjevanje datotek. Razvoj aplikacije je opisan v nadaljevanju.

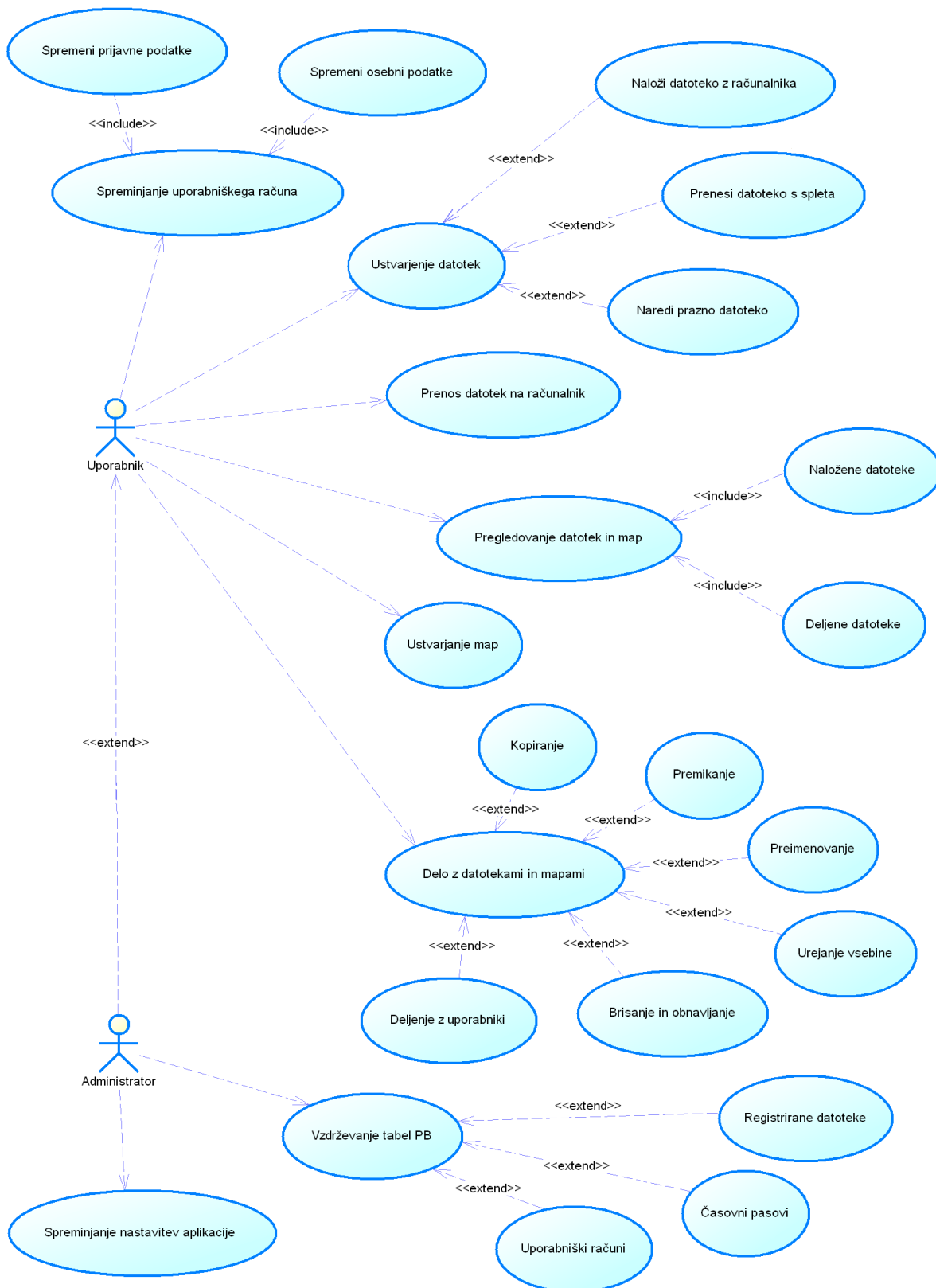
Preden smo se lotili razvoja aplikacije, smo si ogledali obstoječe rešitve za shranjevanje in določili funkcionalnosti. Če bi želeli implementirati vse funkcionalnosti, ki jih nudijo ponudniki tovrstnih storitev, bi bil razvoj preobsežen. V našo aplikacijo smo vključili naslednje funkcionalnosti:

- nalaganje datotek v oblak in prenos iz oblaka na računalnik,
- prenos datotek neposredno s spletnega mesta v oblak,
- ustvarjanje in urejanje tekstovnih datotek na strežniku,
- možnost deljenja datotek z drugimi uporabniki storitve,
- omogočen prenos datoteke komurkoli na spletu,
- funkcije za delo z datotekami: kopiranje, premikanje, preimenovanje, brisanje,
- možnost ogleda podrobnosti o datotekah in mapah,
- iskanje shranjenih datotek in map glede na podane kriterije,
- prilagajanje nastavitev uporabniškega računa,
- pregled pomoči za uporabo aplikacije.

Realizacija funkcionalnosti naše aplikacije je natančneje predstavljena v poglavju Dodatek.

Po končanem razvoju smo aplikacijo namestili na platformo (PaaS), zato jo lahko uporabljamo kdorkoli. Prednost spletnih aplikacij v primerjavi z namiznimi aplikacijami je tudi neodvisnost od programske opreme računalnika ali mobilne naprave, kjer aplikacijo uporabljamo. Potrebujemo le dostop do spleta in odjemalca – spletni brskalnik, s pomočjo katerega uporabljamo aplikacijo.

Diagram primerov uporabe (Slika 4) prikazuje vse funkcionalnosti aplikacije. Predpogoj za uporabo funkcionalnosti je uspešna avtentikacija v spletno aplikacijo (ni prikazano na diagramu). Uporabniki morajo v ta namen opraviti še registracijo, s čimer pridobijo uporabniški račun.



Slika 4: Diagram primerov uporabe aplikacije.

4.2 Predstavitev aplikacije

V nadaljevanju je predstavljen postopek uporabe spletne aplikacije. Ker se aplikacija nahaja na spletu, uporabnik potrebuje le spletni brskalnik, kamor vpiše URL naslov aplikacije.

Registracija, prijava in obnovitev računa

Pred prvo uporabo aplikacije se mora uporabnik najprej registrirati oziroma pridobiti svoj uporabniški račun. Po uspešni registraciji je uporabniku poslana elektronska pošta z navodili za potrditev uporabniškega računa.

Po opravljeni registraciji se lahko uporabnik prijavi in uporablja funkcionalnosti aplikacije. Čeprav ima ogrodje ASP.NET že vgrajen sistem za avtentikacijo, smo zaradi ustreznosti razvili svojega.



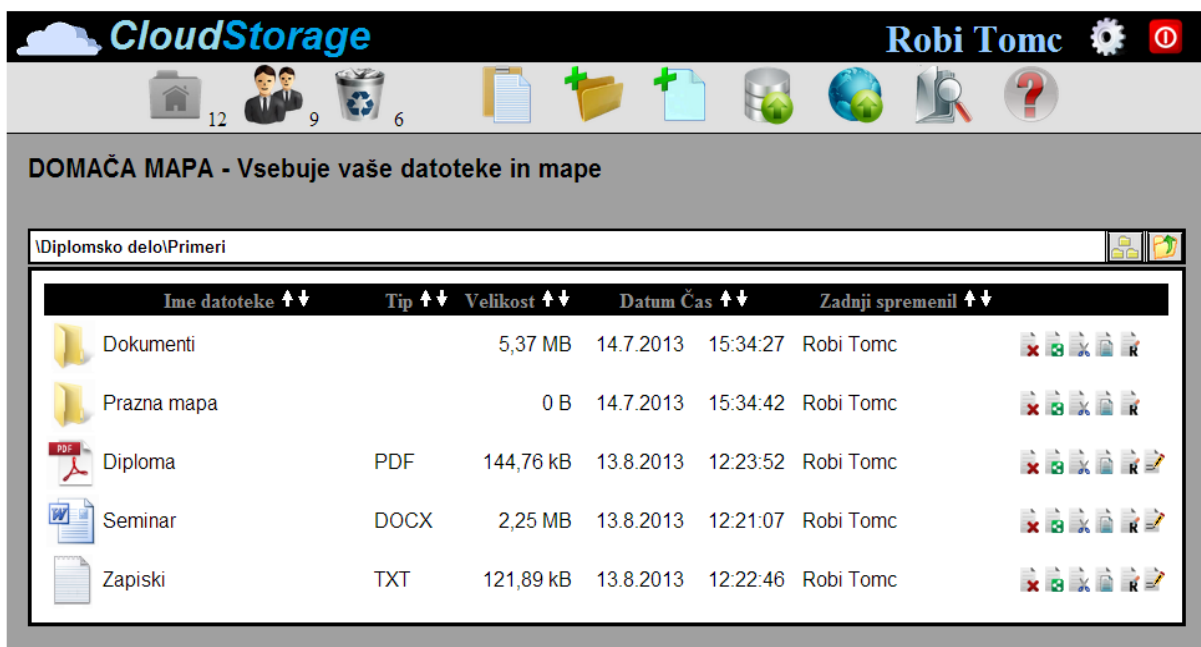
Slika 5: Prijavna stran spletne aplikacije.

Če se uporabnik ne more prijaviti, ima na voljo obrazec za obnovitev uporabniškega računa. Vpisati mora le svoje uporabniško ime ali elektronsko pošto in izbrati vrsto težave. Sistem v podatkovni bazi poišče podatke o uporabniku in mu po elektronski pošti pošlje nadaljnja navodila. Ta funkcionalnost je učinkovita v primerih, ko:

- potrditvena pošta ni prispela ali je bila izbrisana,
- je uporabnik pozabil svoje prijavnne podatke,
- je uporabniški račun začasno blokiran (zamrznitev).

Delo z uporabniškim vmesnikom

Po uspešni prijavi je uporabnik preusmerjen na t. i. domačo stran, kjer lahko pregleduje svoje naložene datoteke in ustvarjene mape. Uporabniški vmesnik (Slika 6) je sestavljen iz brskalnika datotek in gumbov za delo s spletno aplikacijo.



Slika 6: Uporabniški vmesnik z brskalnikom datotek.

Pri izdelavi uporabniškega vmesnika smo upoštevali pogosto postavitve elementov in ustrezno izbiro ikon – če je uporabnik navajen na postavitve elementov, se lažje znajde v novem vmesniku. Za lažjo uporabo smo gumbom v meniju dodali še namige. V desnem zgornjem kotu je prikazan prijavljeni uporabnik; tukaj najdemo tudi gumba za odjavo iz aplikacije ter upravljanje in pregled uporabniškega računa.

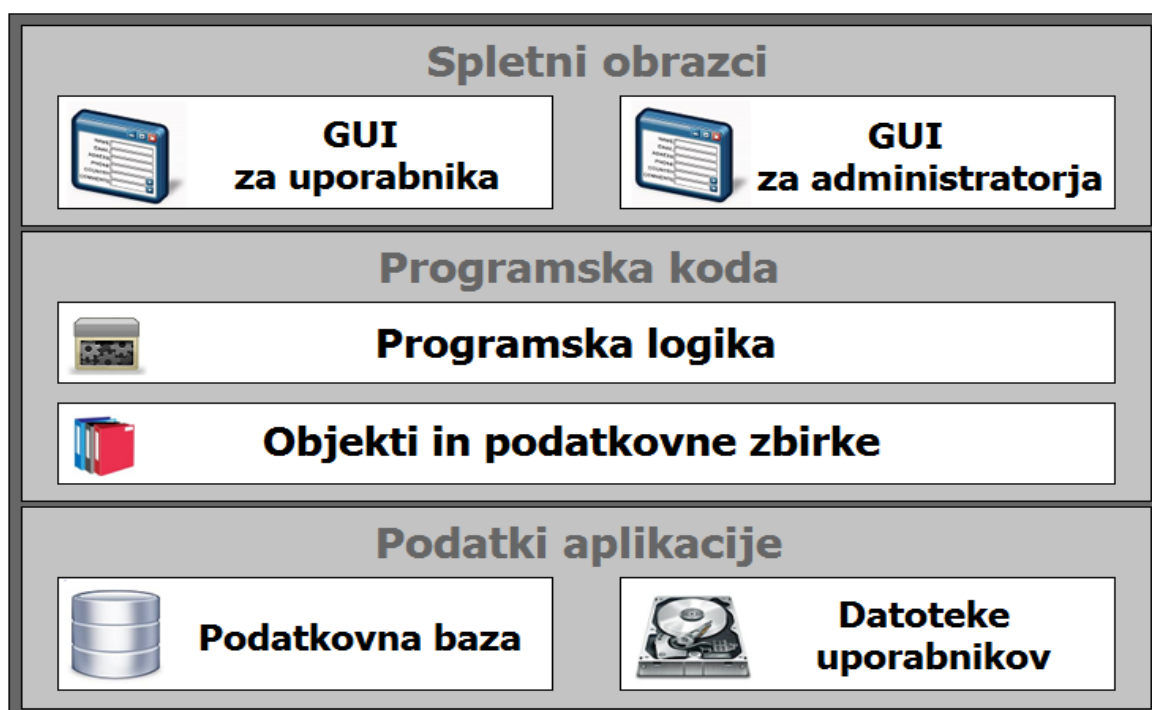
Zgornja vrstica uporabniškega vmesnika vsebuje gumbe za delo s spletno aplikacijo (prikaz naloženih datotek, izbrisanih datotek, ustvarjanje datotek in map, iskanje datotek itd.). Številke poleg prvih treh gumbov pomenijo število elementov v mapi; uporabnik – tako uporabnik lažje sledi spremembam.

Za prenos datoteke na računalnik mora uporabnik klikniti na njeno ime. Prenos datotek, ki so v mapi Koš, ni možen, saj so te izbrisane. Klik na ikono datoteke ali mape prikaže podrobnosti. Če pa uporabnik klikne na ime in priimek drugega uporabnika, bo preusmerjen na njegovo spletno stran. Gumbi poleg imena elementa so namenjeni delu z datoteko ali mapo in so v nekaterih primerih lahko skriti; za delo z datoteko uporabnik potrebuje pravice in če le-teh nima, gumbi niso prikazani.

4.3 Zgradba aplikacije

Spletna aplikacija je sestavljena iz spletnih obrazcev, programske kode in podatkovne baze. Za vsako funkcionalnost je ustvarjen spletni obrazec, ki določa vsebino spletne strani. Vsakemu spletnemu obrazcu pripada datoteka s programsko kodo, ki se izvrši ob ustreznemu dogodku (klik na gumb, slika ipd.).

Za obdelavo podatkov med izvajanjem aplikacije skrbijo podatkovne zbirke objektov, kjer so shranjeni podatki iz podatkovne baze. Za trajno shranjevanje podatkov o datotekah potrebujemo še podatkovno bazo. Zgradbo spletne aplikacije prikazuje Slika 7.



Slika 7: Zgradba spletne aplikacije.

4.4 Predstavitev uporabljenih tehnologij

Spletno aplikacijo smo razvili z ogrodjem ASP.NET. Programska koda v ozadju je napisana v programskem jeziku C#, nekatere funkcije pa v skriptnem jeziku JavaScript. Aplikacija podatke shranjuje v podatkovni bazi, ki jo upravljamo s pomočjo povpraševalnega jezika SQL. Za pisanje in prevajanje programske kode smo uporabili okolje Microsoft Visual Studio 2010, za namestitev podatkovne baze pa Microsoft SQL Server Management Studio 2008. Poleg programiranja je bilo treba nekatere grafične elemente posebej oblikovati.

Microsoft Visual Studio

Microsoft Visual Studio je razvojno okolje (angl. IDE – Integrated Development Environment) za izdelavo programske opreme. Razvijamo lahko različne vrste aplikacij v različnih programskih jezikih: Visual C in C++, Visual C#, Visual F#, Visual J#, Visual Basic. Za razvoj spletnih aplikacij so integrirani še jeziki, kot so JavaScript, ASP, HTML in CSS. Programsko opremo lahko razvijamo le za Microsoftove platforme: Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Microsoft Silverlight.

Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio je programska oprema, ki se uporablja za izdelavo in administracijo podatkovnih baz. Omogoča povezovanje na lokalni in oddaljeni SQL strežnik. Za namestitev potrebujemo ogrodje .NET in ustrezno verzijo SQL Serverja (vsaj SQL Server 2005 ali novejši). Čeprav lahko podatkovne baze oblikujemo tudi v Visual Studiu, prinaša Management Studio številna orodja za razvoj.

Git in GitHub

Git je razvojno orodje in storitev, ki se v razvoju programske opreme uporablja za upravljanje, združevanje izvorne kode ter sinhronizacijo z zunanjimi skladišči oziroma repozitoriji (angl. repository). Orodje je primerno tudi za delo na projektih. Na operacijskih sistemih Linux/Unix se Git namesti ob operacijskem sistemu, na sistemih Microsoft Windows pa ga moramo namestiti sami.

Uporabniški vmesnik je na voljo v več verzijah (GitHub, GitBox ipd.), lahko pa ga uporabljamo tudi preko ukazne vrstice (Git Shell, Git Bash ipd.). Za namestitev naše aplikacije smo uporabili programsko opremo GitHub.

NET in ASP

.NET je Microsoftova platforma za razvoj in poganjanje programske opreme. Omogoča razvoj konzolnih, spletnih ter namiznih Windows aplikacij. Obstaja več različnih verzij ogrodja .NET, zato ob razvoju aplikacij lahko določimo ciljno verzijo.

ASP je spletni skriptni jezik, namenjen razvoju aktivnih spletnih strani (angl. Active Server Pages). Temelji na ogrodju .NET. Po zgradbi gre za tekstovni dokument, ki ima določeno

sintakso. Pisanje ASP kode spominja na HTML, le da tukaj uporabljamo drugačne oznake. Ko spletni strežnik dobi zahtevo po ASP datoteki, se ta ustrezno interpretira, rezultat pa se pošlje odjemalcu. Na ta način spletni strani omogočimo dostop do podatkovnih baz, sistemskih podatkov in imenikov.

C#

C# (C sharp) je eden od programskih jezikov, ki so integrirani v razvojno okolje Microsoft Visual Studio. Po sintaksi je programski jezik precej podoben svojemu predhodniku, programskemu jeziku C++. Spominja tudi na Java in je prav tako objektno usmerjen. Za razvoj in poganjanje programov, napisanih v C#, potrebujemo ogrodje .NET. Pri razvoju spletnih aplikacij ASP.NET se jezik C# uporablja v ozadju. To pomeni, da se programska koda izvede ob določenemu dogodku, na primer ob kliku na gumb.

SQL in SQL Server

SQL je strukturirani povpraševalni jezik za delo s podatkovnimi bazami. Podatkovna baza se lahko nahaja kjerkoli v spletu na podatkovnem strežniku. To omogoča, da so podatki vedno na istem mestu, ne glede na to, kje jih bomo uporabili. Za povezovanje z bazo potrebujemo naslov strežnika, uporabniško ime in geslo.

SQL Server je strežnik za upravljanje podatkovnih baz. Med razvojem naše aplikacije smo uporabljali MS-SQL Server. Strežnik oziroma podatkovno bazo lahko uporabljamo tudi v drugih aplikacijah.

HTML

HTML je hipertekstovni označevalni jezik za izdelavo spletni strani. Omogoča samo izdelavo spletnih elementov (slik, oznak, povezav, vnosnih polj, gumbov), ne pa tudi njihovo oblikovanje. Ko spletni strežnik dobi zahtevo po datoteki, se ta pošlje odjemalcu in se v brskalniku ustrezno prikaže. Jezik HTML je vgrajen v vsak spletni brskalnik, vendar kljub temu moramo paziti katere oznake uporabljamo. Če oznaka ni ustrezno definirana, se lahko zgodi, da bo HTML gradnik v nekem drugem brskalniku napačno prikazan.

CSS

CSS je označevalni jezik, ki se uporablja za oblikovanje HTML gradnikov ter določanje njihove pozicije glede na spletno stran. Jezik CSS je vgrajen v spletni brskalnik. Prednost uporabe CSS je ločevanje med obliko in vsebino. Spletno stran bi sicer lahko prikazali tudi brez uporabe označevalnega jezika, vendar barve ne bi bile uporabljene, postavitev gradnikov pa bi bila precej drugačna. Z uporabo CSS lahko za isto spletno stran uporabljamo različne predloge, kar omogoča hitrejše urejanje in večjo preglednost.

JavaScript

JavaScript je skriptni jezik, ki omogoča izvajanje preprostih ukazov na strani odjemalca. Posamezne programske funkcije lahko povežemo s HTML gradniki. To nam omogoča, da se ob nekem dogodku (na primer klik na gradnik) izvede ustrezna programska koda. Na ta način spletnim stranem dodamo interaktivnost.

JavaScript ni Java, kakor nekateri sklepajo. Java je programski jezik in potrebuje ustrezen prevajalnik, JavaScript pa je skriptni jezik, vgrajen v spletni brskalnik, zato ne potrebujemo prevajalnika in posebnega spletnega strežnika.

Piškotki

Piškotki (angl. cookies) so kratke tekstovne datoteke, ki jih preko spletnega brskalnika ustvari spletni strežnik. Na ta način se shranijo določene informacije, da nam jih ob naslednjem obisku spletne strani ni treba ponovno nastavljanje. Poleg vsebine lahko piškotkom določimo tudi čas izteka veljavnosti (ob določenem času, ob zaprtju brskalnika itd). Piškotke lahko uporabljamo za zagotavljanje boljše funkcionalnosti, v nekaterih primerih pa brez njih ne moremo (seja – prijava v aplikacijo). Ker uporaba piškotkov posega v zasebnost uporabnikov, morajo biti uporabniki ob obisku spletne strani obveščeni o uporabi piškotkov.

Objekti in podatkovne zbirke

Objekt je programski gradnik, sestavljen iz več spremenljivk oziroma atributov in metod. Pri shranjevanju enostavnih podatkov uporabljamo osnovne podatkovne tipe (celo število, decimalno število, znak, logična vrednost itd.), pri delu z objekti pa imamo opravka z različnimi vrstami podatkov, datoteko določa več atributov (npr. ime, pot, velikost itd).

Podatkovna zbirka (angl. collection) je struktura za shranjevanje objektov. Čeprav se morda zdi tako, to ni podatkovna baza. Podatkovne baze trajno hranijo podatke, zbirke pa obstajajo le znotraj programa. Do posameznega objekta v podatkovni zbirki dostopamo preko t. i. indeksov. Prednost uporabe podatkovnih zbirk je tudi neomejena količina elementov, ki jih lahko shranimo (v primerjavi s tabelami) ter nabor funkcij za delo z elementi. Nekaj primerov podatkovnih zbirk: seznam (angl. list), vrsta (angl. queue), sklad (angl. stack).

4.5 Podatkovna baza

Za shranjevanje podatkov uporablja aplikacija podatkovno bazo (Slika 8). V njej shranjujemo tako podatke uporabnikov kot tudi podatke, ki so potrebni za delovanje aplikacije. V nadaljevanju je opisan pomen posameznih tabel.

Users (uporabniki): Za vsakega uporabnika imamo shranjene njegove podatke in podatke o uporabniškem računu (kapaciteta, datum spremembe), prijavnne podatke (uporabniško ime, MD5 zgoščena vrednost gesla) in časovne nastavitve (časovni pas). Ko se uporabnik registrira, se v tabelo doda nov vnos, ki predstavlja uporabnika.

Files (datoteke): Tabela vsebuje opis vseh datotek, ki so jih uporabniki shranili. Shranjujemo ime datoteke, pot, lastnika, velikost itd. Nekatere podatke bi lahko dobili z uporabo funkcij za dostop do datotečnega sistema (datoteke), vendar je dostop do podatkovne baze hitrejši, zato ob prenosu datoteke shranimo njene podatke v tabelo.

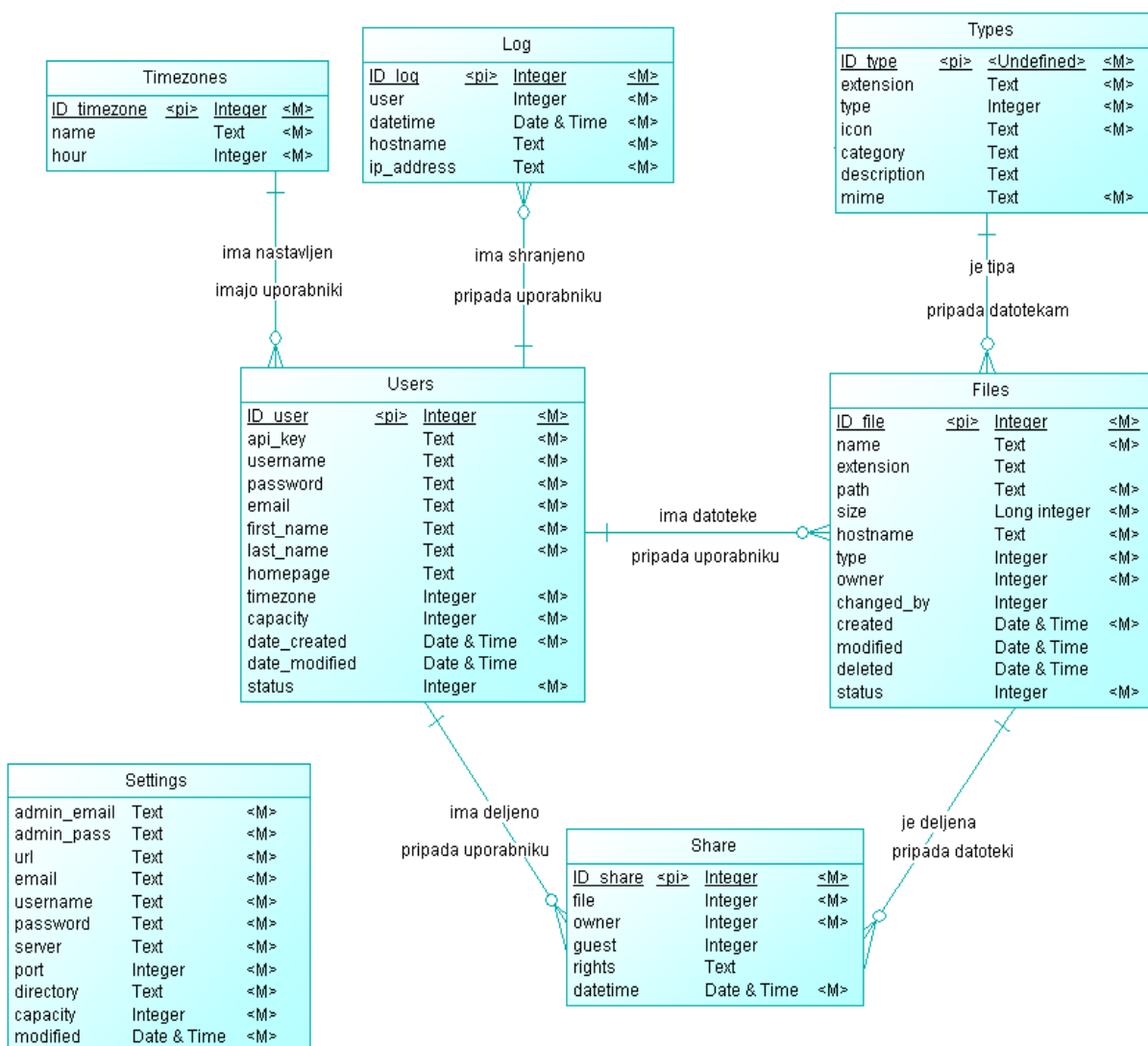
Share (deljene datoteke): Ko uporabnik deli svojo datoteko z drugim uporabnikom, se v tabelo doda vnos s podatki o delitvi (kdo je delil, katero datoteko in kdaj, kakšne pravice ima uporabnik do datoteke). Na ta način lahko uporabnik deli datoteko v več uporabniki, vendar ima vsak od njih drugačne pravice.

Types (tipi datotek): V tabeli so shranjeni podatki o različnih tipih datotek (TXT, BMP, ZIP, EXE itd.). Podatki, ki jih shranjujemo, so pomembni tako za predstavitev (ikona, opis) kot tudi za prenos datoteke na računalnik (MIME podatki). Administrator lahko kadarkoli doda nove tipe datotek ali spremeni obstoječe podatke.

Log (dnevnik prijav): Vsakokrat, ko se uporabnik prijavi v aplikacijo, se v tabelo doda vnos s podatki o prijavi (datum, čas, IP naslov, ime računalnika). Uporabnik ima tako pregled nad uporabo aplikacije – lahko si ogleda datum zadnje prijave ali prenese dnevniško datoteko, kjer so navedeni podatki o prijavih. Podatke o prijavih lahko uporabnik tudi izbriše.

Timezones (časovni pasovi): Tabela predstavlja seznam časovnih pasov oziroma regij, med katerimi lahko izbira uporabnik. Za vsako regijo imamo shranjeno njeno ime in ustrezen časovni odmik, ki ga uporabljamo za pravilen prikaz datuma in časa. Uporabnik mora v nastavitvah izbrati svojo regijo. Administrator lahko kadarkoli doda nove regije ali spremeni obstoječe podatke.

Settings (nastavitve): Podatke v tej tabeli lahko spreminja le administrator, saj so potrebni za pravilno delovanje aplikacije. V tabeli shranjujemo privzeto kapaciteto uporabniških računov, ime mape za shranjevanje datotek, podatke za pošiljanje elektronske pošte in prijavnne podatke za administratorja.

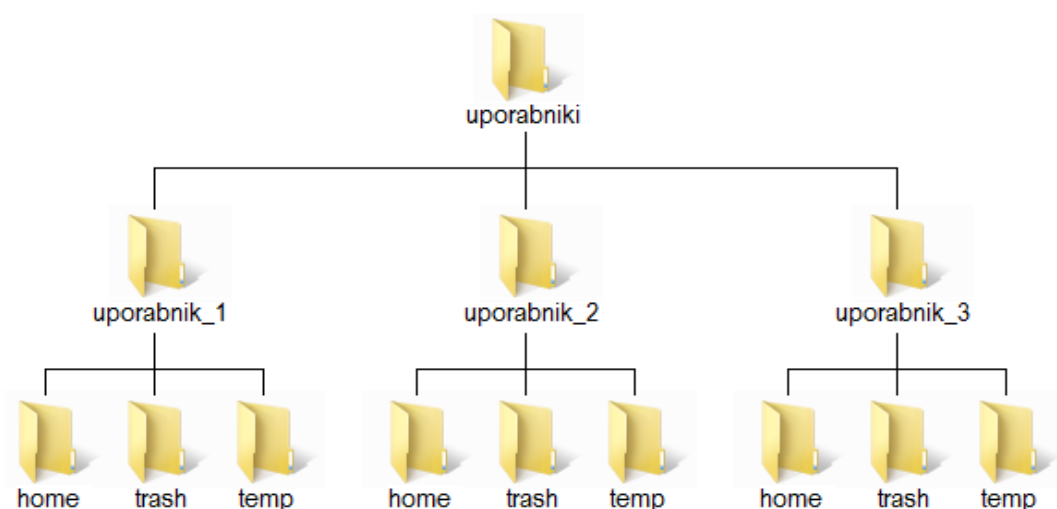


Slika 8: Konceptualni model podatkovne baze.

4.6 Struktura map za shranjevanje datotek

Za shranjevanje datotek uporabnikov ima aplikacija definirano strukturo map (Slika 9). Na najvišjem nivoju se nahaja mapa, znotraj katere je se ustvarjajo mape uporabnikov. Ime oziroma pot do mape mora skrbnik spletne aplikacije nastaviti ob namestitvi v oblak. Čeprav je mapa del neke druge poti, je v programu predstavljena kot korenski imenik (angl. root).

Na drugem nivoju so mape uporabnikov. Ko se uporabnik registrira v storitev, se ustvari mapa, katere ime je enako uporabniškemu imenu uporabnika. Znotraj mape vsakega uporabnika najdemo podmape, kamor se shranjujejo datoteke (tretji nivo).



Slika 9: Drevesna struktura map uporabnikov.

Pomen posameznih map:

- »home«: Vsebuje datoteke uporabnika. Uporabnik lahko znotraj mape ustvari še dodatne mape za shranjevanje, zato se struktura map nadaljuje;
- »trash«: Vsebuje datoteke in mape, ki jih je uporabnik izbrisal (implementacija Koša). Teh datotek uporabnik ne more prenesti ali spreminjati, saj so pripravljene za brisanje. Lahko jih le trajno izbriše ali obnovi;
- »temp«: Vsebuječasne datoteke. Tukaj aplikacija ustvari datoteke, ki so potrebne za pravilno delovanje storitve. Po končani uporabi se datoteke izbrišejo.

S stališča uporabnika so navedene mape nevidne, saj ne pozna njihovih imen ali lege v strukturi. Do shranjenih datotek lahko dostopa preko spletnih obrazcev aplikacije. Navedenih map uporabnik ne more izbrisati. Mapa posameznega uporabnika se odstrani le ob zaprtju uporabniškega računa.

4.7 Izzivi in zapleti med razvojem

Med razvojem spletne aplikacije je večkrat prišlo do manjših težav, ki so bile posledica netočnega načrtovanja oziroma neupoštevanja vseh možnosti. Taki zapleti so hkrati predstavljali izziv za nadaljnji razvoj. V nadaljevanju je predstavljenih nekaj primerov.

Prenos podatkov med spletnimi obrazci

V nekaterih primerih se podatki med obrazci prenašajo kot parametri v URL naslovu. Tak pristop je primeren, kadar želi uporabnik nekaj početi z datoteko in potrebuje podatke o ID datoteke in katera operacija naj se izvrši nad datoteko. Problem nastane, ker lahko uporabnik parametre spremeni, zato je pred delom z datoteko potrebno preverjanje veljavnosti in pravic uporabnika.

V nekaterih primerih je podatke treba shraniti za dalj časa: označevanje datoteke za kopiranje, nastavitve sortiranja datotek, shranjevanje odprte mape itd. To lahko dosežemo z uporabo piškotkov. Prednost tega pristopa je, da so podatki vidni v vseh spletnih obrazcih in jih uporabnik ne more spremeniti.

Prikazovanje datuma in časa

Ko smo v oblak namestili in testirali prvo delujočo verzijo, smo opazili, da se prikazujeta napačen čas in datum, čeprav je bilo tekom razvoja vse pravilno prikazano. Napaka je posledica jezikovnih in časovnih nastavitvev na strani strežnika. Na isti problem bi naleteli, če bi imel odjemalec nastavljen drugi časovni pas kot strežnik.

Težavo smo rešili tako, da smo implementirali nastavljanje časovnega pasu. Uporabnik si po opravljeni registraciji nastavi ustrezen časovni pas. Podatki o časovnih nastavitvah so v posebni tabeli v podatkovni bazi, pri izpisu moramo upoštevati le nastavitvev ter dodati ali odvzeti ustrezen časovni odmik. Za pravilen izpis morajo biti v tabelo vpisani točni podatki, za kar mora poskrbeti administrator.

Poimenovanje datotek in map

Kot smo že omenili, imamo lahko v mapah samo eno datoteko ali podmapo z določenim imenom in končnico. Pri kopiranju, premikanju, nalaganju in obnavljanju zato lahko pride do

duplikatov – novonastala datoteka ima lahko ime in končnico, ki že obstajata. V tem primeru sta možni dve rešitvi:

- novejšo datoteko preimenujemo in tako obdržimo obe datoteki,
- prepíšemo starejšo datoteko in obdržimo samo novejšo datoteko.

Ker želimo preprečiti nenamerno izgubo datotek, smo implementirali prvo možnost. Preden uporabnik v mapo vstavi nov element, se ta po potrebi preimenuje.

Kopiranje in premikanje mape

Programski jezik C# ima funkcije za kopiranje in premikanje map že vgrajene, zato se ta procesa morda ne zdita nič posebnega. Problem pri uporabi obstoječih funkcij je predvsem v tem, ker se moramo držati določenih omejitev:

- Popravek ali vnos v podatkovno bazo: Vsaka datoteka ima v podatkovni bazi shranjene lastnosti. Če datoteko kopiramo, moramo podvojiti tudi njen vnos v bazi. Pri premikanju datoteke moramo v bazi posodobiti njeno absolutno pot.
- Mapa lahko vsebuje izbrisane datoteke: Pri kopiranju mape moramo upoštevati, da morda vsebuje datoteke ali podmape, ki niso dokončno izbrisane. Take datoteke so še vedno shranjene na disku, pravimo, da so v košu. Teh datotek ni treba kopirati, saj bi tako le porabili prostor.
- Upoštevanje omejitve glede kapacitete: Shranjenih imamo lahko le toliko datotek, kolikor znaša kapaciteta uporabniškega računa. Pri kopiranju map se lahko zgodi, da ni dovolj prostora. V takem primeru bi morali skopirati le nekatere datoteke.

Da bi upoštevali vse navedene omejitve, smo morali realizirati lastno funkcijo za kopiranje. Postopek delovanja je sledeč: Najprej ustvarimo vse potrebne mape, potem pa kopiramo vsako datoteko posebej. Pred kopiranjem moramo preveriti ali datoteka res obstaja in ali je dovolj prostora. Po kopiranju datoteke pa moramo njene lastnosti še shraniti v bazo. Enako velja za premikanje map.

Brisanje in obnavljanje – uporaba mape Koš

Tudi funkcije za brisanje datotek in map so že napisane, vendar datotek ne smemo takoj izbrisati, ampak jih moramo najprej prestaviti v mapo Koš. Prav zaradi tega uporaba obstoječih funkcij ni primerna. Izdelati smo morali lastne funkcije in pri tem upoštevati naslednje zahteve:

- Ustrezna predstavitev datotek: Če v neki mapi izbrišemo dve datoteki in na to še celotno mapo, morajo biti v Košu predstavljeni tri ločeni elementi. Če iz Koša izbrišemo mapo, morata datoteki ostati, čeprav sta bili pred izbrisom del mape.
- V Košu je lahko več elementov z enakim imenom in končnico: Za razliko od drugih map, Koš lahko vsebuje več datotek ali map z enakim imenom in končnico. Do tega pride, če iz različnih map izbrišemo enako poimenovane datoteke.
- Pravilna obnovitev datoteke ali mape: Ko element obnovimo, ga je treba prestaviti na mesto, kjer je bil pred izbrisom. Pri tem lahko nastane problem, če je del poti izbrisan.

Za pravilno delovanje mora naša funkcija za brisanje delovati na naslednji način:

- Če brišemo mapo, moramo vedeti, katera mapa je bila neposredno izbrisana ter katere datoteke in mape so izbrisane, ker so bile v tej mapi. V ta namen ima vsaka datoteka definiran atribut »status«, ki opisuje stanje datoteke.
- Izbrisani element je treba ločiti od mape, kjer se nahaja. Za najbolj ugodno rešitev se je izkazalo spreminjanje atributa »path« (pot). Na začetek poti dodamo ID elementa; zdi se, da je element postal del neke druge mape, zato ob brskanju ni prikazan.
- Izbrisani element moramo še fizično prestaviti drugam. V ta namen ima vsak uporabnik dodeljeno mapo »trash« (smeti), kamor se odlagajo izbrisani elementi. Upoštevati moramo, da je lahko v Košu več elementov poimenovanih enako, zato moramo element še preimenovati – na začetek imena dodamo ID elementa. Četudi je preostali del imena enak, se ID razlikuje.

Obnavljanje izbrisanih datotek je preprosto, saj imamo še vedno vse podatke, le popraviti jih moramo. V podatkovni bazi atributu »path« izbrišemo del, ki predstavlja ID; enako naredimo z imenom datoteke. Preverimo, ali zahtevana pot obstaja, in po potrebi ustvarimo strukturo map. Nato element prestavimo tja, kjer je bil pred izbrisom.

Izpis pomoči uporabnikom

V spletno aplikacijo smo vgradili pomoč uporabnikom. Predvideli smo morebitna vprašanja, ki bi jih lahko imel uporabnik, in navedli rešitve. Celotno vsebino pomoči bi sicer lahko shranili v programsko kodo, vendar bi bilo to nepregledno, poleg tega bi morali v primeru vzdrževanja program spreminjati.

Namesto tega smo ustvarili več tekstovnih datotek in vanje shranili vsebino pomoči. Kadarkoli uporabnik odpre eno od poglavij pomoči, program prebere ustrezno datoteko in jo izpiše na spletno stran.

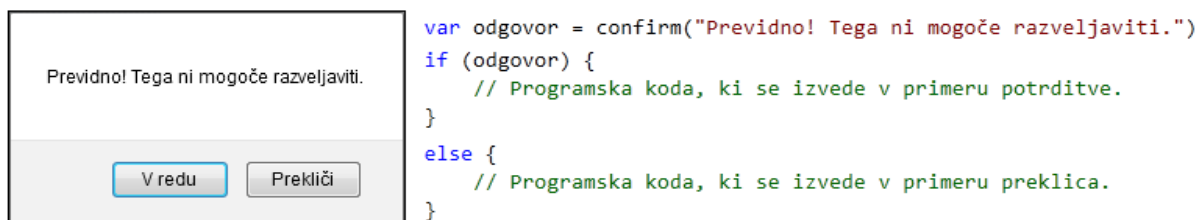
4.8 Možne izboljšave

Čeprav je bila spletna aplikacija testirana in izpolnjuje svoj namen, se pojavljajo nove ideje za nadgradnjo. V nadaljevanju je opisanih nekaj idej, ki bi jih bilo dobro upoštevati.

Zaščita pred nenamernimi dejanji

Lahko se zgodi, da uporabnik pomotoma klikne nek gumb in sproži dejanje, ki ga morda ni mogoče razveljaviti. Za primer vzemimo postopek brisanja datotek. Uporabnik klikne ikono za brisanje in datoteka se izbriše brez opozorila oziroma zahteve po nadaljevanju.

Taka nenamerna dejanja bi lahko preprečili tako, da se pred brisanjem uporabniku prikaže dialog, v katerem bi uporabnik potrdil svoj namen. Za implementacijo bi lahko uporabili JS funkcijo `confirm()`, ki bi jo povezali s HTML gradnikom.



Slika 10: Primer opozorila in pripadajoče programske kode.

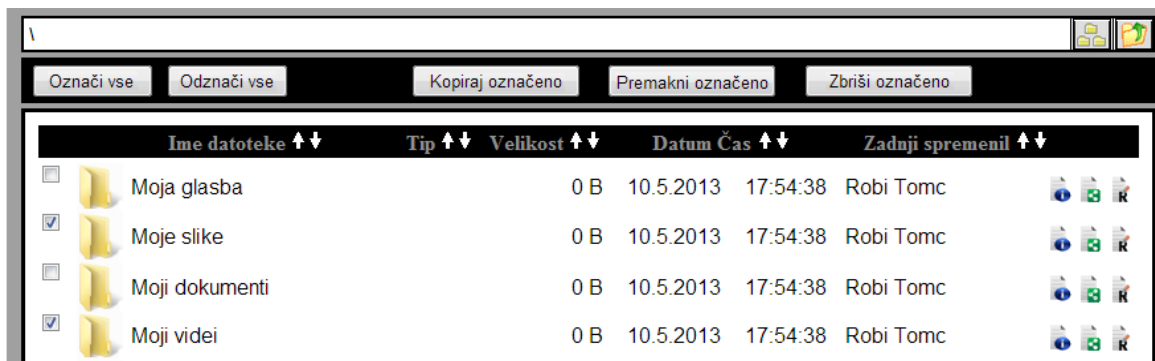
Izboljšano iskanje datotek

Kadar ima uporabnik več datotek, se pojavi problem pri iskanju določene datoteke. Spletna aplikacija ima sicer vgrajen vmesnik za iskanje, vendar bi ga bilo možno izboljšati. Dodamo lahko več atributov za iskanje (kdaj je bila datoteka nazadnje spremenjena, kdo je spremenil, iz katerega računalnika je bila naložena itd.). Pri tekstovnih datotekah bi lahko dodali tudi možnost iskanja po vsebini. Sistem bi moral vsako datoteko prebrati in preveriti, ali vsebina vsebuje podane ključne besede. Če so ključne besede najdene, se ime datoteke izpiše.

Operacije nad več datotekami hkrati

Problem se pojavi tudi pri delu z datotekami. Če bi hoteli skopirati več datotek, bi morali označiti vsako posebej, poiskati mapo in datoteko prilepiti. Postopek je lahko dolgotrajen in zahteva veliko potez. Možna izboljšava bi bila možnost označevanja datotek. Poleg vsake datoteke bi bilo polje za izbor (angl. checkbox). Ko bi uporabnik izvedel neko operacijo, bi ta

zajela vse označene datoteke. Prilagoditi je treba tudi uporabniški vmesnik in programsko logiko. Slika 11 prikazuje uporabniški vmesnik, ki bi izbor omogočal.



Slika 11: Prilagojen uporabniški vmesnik za izbiro datotek.

Izboljšan prikaz podrobnosti datotek

Pomanjkljivost pri trenutni verziji aplikacije je prikaz podatkov o datotekah in mapah. Na seznamu je prikazanih le nekaj osnovnih podatkov: ime, končnica, velikost, datum in čas spremembe, kdo jo je zadnji spremenil. Za več podatkov mora uporabnik odpreti obrazec za ogled podrobnosti o datoteki.

Za boljši pregled nad arhivom bi lahko implementirali izbiro prikazanih podatkov. V obrazcu z nastavitvami bi uporabnik izbral podatke, ki naj se prikazujejo ob datoteki. Za nadgradnjo bi morali spremeniti naslednje:

- Podatkovna baza: V tabelo uporabniških računov bi morali dodati dodaten atribut, kjer bi za vsakega uporabnika hranili nastavitvev.
- Obrazec za brskanje: Preden prikažemo vsebino, bi morali iz podatkovne baze prebrati nastavitvev in temu ustrezno prilagoditi prikaz. Ker vsebino mape izpisujemo v HTML tabelo, bi morali izpisati ustrezno število stolpcev, nastaviti ustrezno poravnavo in izpisati podatek.
- Obrazec za nastavitve: Implementirati moramo možnost izbora podatkov. Uporabimo lahko polja za izbor (angl. checkbox), kjer naštejemo vse podatke. Ko uporabnik izbere podatke, se spremembe shranijo v podatkovno bazo.

Delo s stisnjenimi datotekami

Nekateri ponudniki shranjevanja ponujajo to funkcionalnost; gre za to, da več datotek ali map shranimo v eno samo datoteko (ZIP ali RAR arhiv). Recimo, da bi želeli na računalnik prenesti več datotek – ob uporabi trenutne verzije bi morali prenesti vsako datoteko posebej,

izboljšava pa bi omogočila, da celotno mapo shranimo v eno datoteko in prenesemo le-to. Podoben problem se pojavi pri nalaganju na strežnik. Namesto več datotek prenesemo eno samo ZIP datoteko, ki jo potem razpakiramo.

Sledenje verzijam dokumentov

Velikokrat se zgodi, da moramo datoteko večkrat spremeniti. Večina storitev omogoča sledenje verzijam, kar pomeni, da lahko pregledujemo spremembe in jih kadarkoli razveljavimo. Naša aplikacija te funkcionalnosti nima. Implementacija bi bila možna na naslednji način: Podatkovni bazi dodamo tabelo »Verzije«, kjer bi shranjevali ime datoteke, pot, velikost, datum in čas spremembe ter tuji ključ za povezavo s tabelo datotek. Pred spreminjanjem datoteke bi bilo treba narediti kopijo ter podatke o datoteki zabeležiti v tabelo verzij. Za shranjevanje različnih verzij bi morali z dodano posebno mapo razširiti še strukturo map uporabnika.

Kriptiranje datotek

Trenutna verzija aplikacije ne omogoča šifriranja shranjenih datotek. V primeru napada so vse naložene datoteke ogrožene, saj si napadalec lahko ogleda vsebino. Za varnost svojih datotek morajo uporabniki poskrbeti sami, tako da svoje datoteke pred nalaganjem šifrirajo z uporabo ustrezne programske opreme.

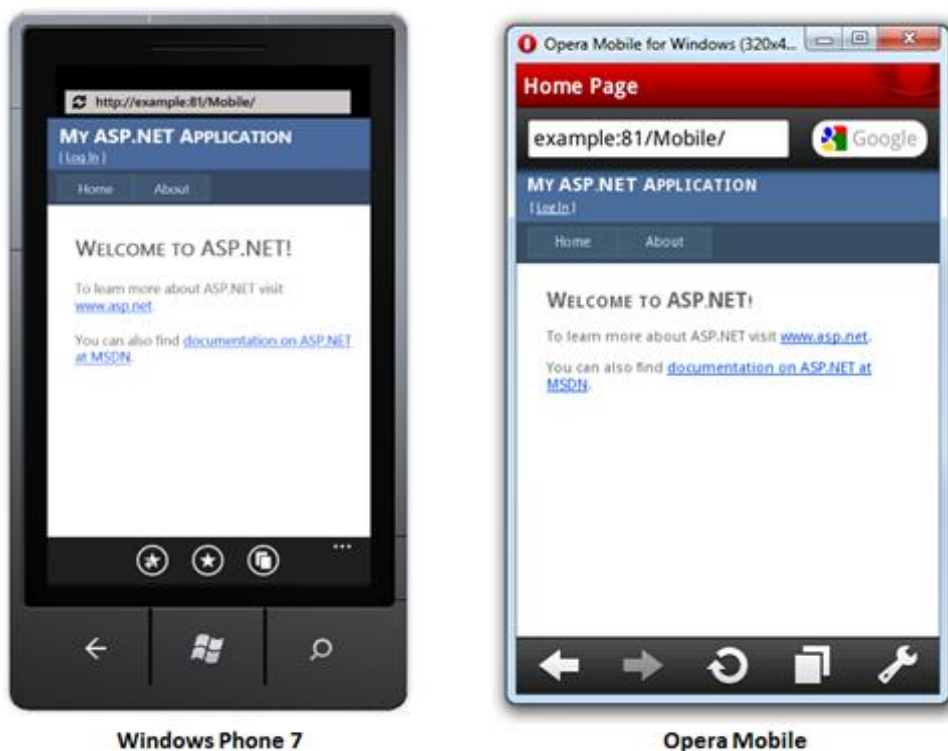
Da bi zagotovili večjo varnost datotek, bi morali vsako naloženo datoteko takoj kriptirati. Ena od možnih rešitev je uporaba vgrajenih C# funkcij za šifriranje z DES algoritmom. Ob registraciji bi vsak uporabnik dobil svoj ključ, s katerim bi se datoteke kriptirale. Pri tem pa naletimo na nove ovire:

- Prenašanje datotek na računalnik: Preden uporabnik datoteko prenese na računalnik, jo je potrebno dešifrirati, saj je sicer neuporabna. Gre za obratno operacijo kot pri shranjevanju datotek.
- Delo z datotekami drugih uporabnikov: Praviloma ima vsak uporabnik svoj ključ za šifriranje. Če bi uporabnik želel svojo datoteko kopirati v mapo drugega uporabnika, je najprej potrebno dešifriranje, nato pa še šifriranje z drugim ključem.
- Pregledovanje tekstovnih datotek: Če bi radi prikazali vsebino tekstovne datoteke, bi to morali najprej dešifrirati.

V vseh treh opisanih situacijah bi morali ustvarjati začasne datoteke. Izboljšava je seveda možna, vendar zahteva veliko sprememb v programu.

Prilagoditev za pametne telefone

Celoten uporabniški vmesnik bi lahko prilagodili tako, da bi aplikacijo lahko uporabljali na pametnih telefonih in mobilnih napravah. Zaslona mobilne naprave je precej manjši od zaslona računalniškega monitorja, zato se uporablja vertikalna postavitev elementov. ASP.NET ima že vgrajene predloge za spletne strani (Slika 12) in MVC aplikacije.



Slika 12: Predloga ASP.NET aplikacije za mobilne naprave.

(vir: http://i1.asp.net/asp.net/images/whitepapers/mobile/image002.png?cdn_id=2013-05-10-001)

Večjezični uporabniški vmesnik

Z implementacijo večjezičnega uporabniškega vmesnika bi lahko aplikacijo uporabljali tudi uporabniki, ki ne govorijo slovenskega jezika. Za vsak jezik bi morali narediti svojo datoteko, kjer bi bile shranjene oznake na gumbih in ostalih gradnikih. Najprej bi bilo treba prilagoditi podatkovno bazo, tako da tabeli uporabnikih računov dodamo atribut »jezik«. Ob naložitvi kateregakoli spletnega obrazca, bi program najprej preveril jezikovno nastavitev, prebral ustrezno datoteko, poiskal prevod in spremenil oznako na gradniku. Za implementacijo bi lahko uporabili podatkovne zbirke, npr. slovarje. Slovarji za razliko od drugih zbirk vrednosti shranjujejo v parih [ključ, vrednost], zato hitro najdemo iskano vrednost.

5 Namestitev spletne aplikacije

5.1 Platforma kot storitev

Izdelano spletno aplikacijo smo namestili na platformo (Platform as a Service – PaaS), tako da je na voljo uporabnikom. Aplikacija, nameščena na platformo, je neodvisna od operacijskega sistema in programske opreme odjemalca, zato jo lahko uporablja kdor koli. Ker različni ponudniki platform ponujajo različne storitve, moramo biti pri izbiri pozorni. Poiskati je bilo treba ponudnika PaaS, ki omogoča:

- podporo za .NET aplikacije (ASP.NET in programski jezik C#),
- dostop do datotečnega sistema (shranjevanje datotek uporabnika),
- uporabo elektronske pošte (pošiljanje potrditvene e-pošte za registracijo),
- delo s podatkovnimi bazami (SQL podatkovni strežnik).

Postopek namestitve spletne aplikacije na platformo je odvisen od ponudnika. Prav tako moramo pred namestitvijo prilagoditi določene nastavitve projekta. Celoten postopek je opisan v nadaljevanju.

5.2 Predstavitev ponudnikov platform

Pred namestitvijo spletne aplikacije na platformo smo morali poiskati primernega ponudnika storitve. Nekaj aktualnih je predstavljenih v nadaljevanju.

Google App Engine

Google App Engine (GAE) je Googlova storitev, ki združuje infrastrukturo, platformo in programsko opremo. Trenutno podpira programske jezike Java (vključno z JVM in Java servleti), Python, Go in PHP [21]. Do svojih aplikacij lahko dostopamo preko lastne domene (nastavimo svoj URL naslov). Aplikacije se izvajajo v t. i. peskovnikih (angl. sandbox), zaradi česar imamo omejen dostop do operacijskega sistema. Storitve se obračunavajo po načinu »pay-per-user«. Za shranjevanje aplikacij ponuja Google naslednje možnosti:

- App Engine Datastore,
- Google Cloud SQL (SQL storitve – za shranjevanje podatkovnih baz),
- Google Cloud Storage (za shranjevanje do 1 TB velikih datotek).

Microsoft Windows Azure

Windows Azure je Microsoftova storitev za postavitev lastne infrastrukture (IaaS) in platforme (PaaS). Podpira več programskih jezikov: ASP.NET, PHP in Node.js. Nameščeno programsko opremo lahko upravljamo s pomočjo orodij FTP, Git ali TFS (Team Foundation Server). Aplikacije delujejo na Windows in Linux strežniški virtualizaciji, zato jih lahko razvijalci prenašajo med sistemoma. Za delo s podatkovnimi bazami je na voljo dodatek SQL Azure, ki temelji na tehnologiji Microsoft SQL Server.

AppHarbor

AppHarbor je platforma za poganjanje .NET programske opreme. Za prenos programske kode se uporablja programska oprema za delo z verzijami (Git, Mercurial ipd.). Če prenesena programska koda ne vsebuje napak, se aplikacija prevede in shrani na strežnik. Za uporabo dodatnih storitev (delo s podatkovnimi bazami, elektronsko pošto, nadzor aplikacije) moramo v nameščeno aplikacijo vključiti ustrezne dodatke (angl. add-ons). Večina dodatkov je brezplačnih, vendar imajo omejeno funkcionalnost, za neomejen dostop pa je potrebno naročilo.

Slabost platforme je, da omogoča le poganjanje aplikacij, ne pa tudi trajnega shranjevanja datotek. Datoteke, ki jih aplikacija ustvari, so le začasne in se izbrišejo vsakokrat, ko aplikacijo ponovno namestimo. Za trajno shranjevanje datotek moramo poiskati drugačno rešitev, npr. shranjevanje pri drugem ponudniku.

Apprenda

Storitev Apprenda uporabnikom in podjetjem omogoča postavitev zasebne platforme. Namestimo lahko neomejeno število aplikacij, ki temeljijo na tehnologiji .NET ali programskem jeziku Java.

Brezplačni uporabniški račun je na voljo vsakomur in ponuja 3,1 GHz procesorske moči, 2 GB delovnega pomnilnika in kapaciteto 100 MB za shranjevanje datotek [22]. Po naročilu lahko katerikoli vir razširimo glede na naše potrebe. Na voljo imamo tudi programsko opremo Apprenda Express, s katero lahko upravljamo svoj oblak.

Uhuru AppCloud

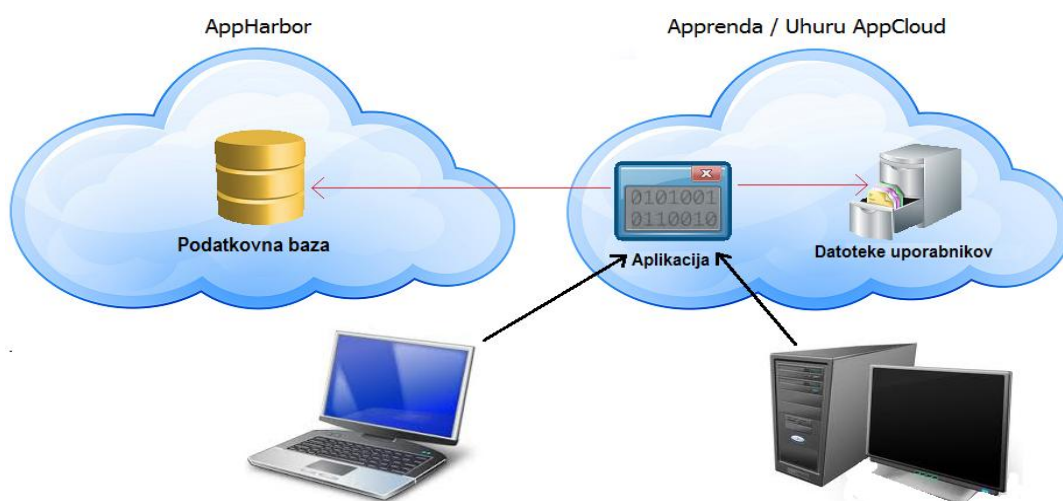
Uhuru AppCloud je storitev, ki omogoča gostovanje spletnih aplikacij, napisanih v eni izmed tehnologij: PHP, Node.js, Java, Scala, Ruby, Erlang, C# .NET. Za nameščanje in upravljanje

nameščenih aplikacij lahko uporabimo programsko opremo Uhuru Cloud Manager, vendar ta deluje le na operacijskem sistemu Microsoft Windows.

Namestimo lahko neomejeno število aplikacij, v vsako pa lahko vključimo še dodatke (angl. add-ons) za delo s podatki. Za gostovanje odprtokodnih aplikacij se uporabljajo Linux operacijski sistemi, medtem ko .NET aplikacije delujejo na operacijskem sistemu Microsoft Windows.

5.3 Namestitev aplikacije

Pri izbiri ponudnika platforme smo se omejili tudi na ceno storitve. Ker večina ponudnikov zahteva naročilo in plačilo uporabe storitve, smo poiskali brezplačno kombinacijo. Namesto enega ponudnika smo izbrali dva (Slika 13), ki skupaj izpolnjujeta vse zahteve za delovanje naše spletne aplikacije. V prvi oblak (ponudnik AppHarbor) smo namestili podatkovno bazo. Za drugi oblak smo uporabili ponudnika Apprenda in Uhuru AppCloud. V njiju smo namestili aplikacijo, hkrati pa v njun datotečni sistem uporabniki shranjujejo svoje datoteke.



Slika 13: Namestitev aplikacije v oblak.

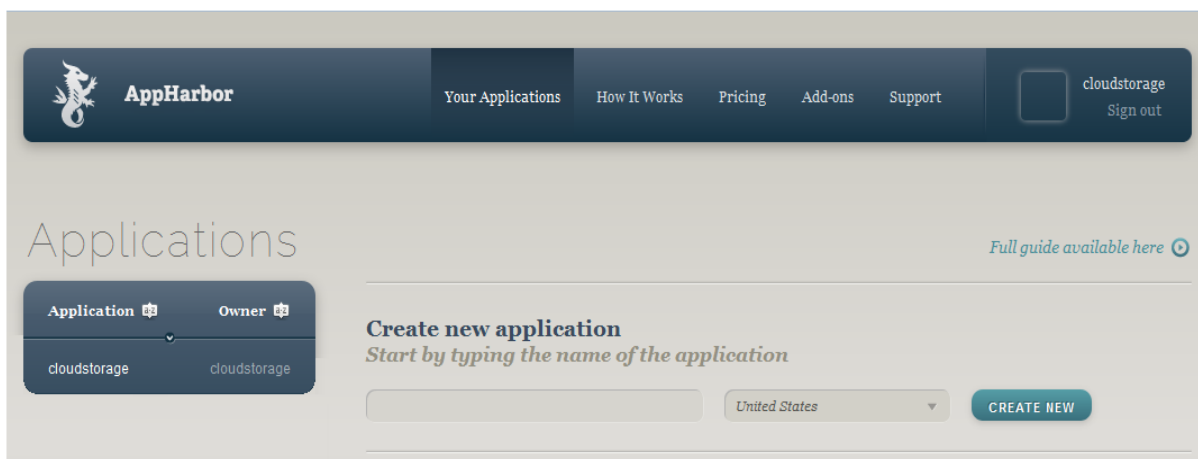
Celoten postopek namestitve lahko razdelimo na 4 ločene dele:

- pridobitev SQL strežnika in elektronske pošte,
- namestitev spletne aplikacije na platformo,
- namestitev podatkovne baze na oddaljeni SQL strežnik,
- konfiguracija in priprava aplikacije za uporabo.

5.3.1 Pridobitev SQL in SMTP strežnika

Kot smo že omenili, aplikacija za delovanje potrebuje podatkovni strežnik in podatke o elektronski pošti. Vse potrebno smo dobili pri ponudniku platforme AppHarbor.

Za uporabo platforme se moramo najprej registrirati. Izpolniti je treba obrazec, kjer si izberemo prijavne podatke. Po končani registraciji se lahko prijavimo in uporabljamo storitev. Prikaže se uporabniški vmesnik, kjer lahko urejamo svoje aplikacije. Tukaj smo izpolnili obrazec za ustvarjanje nove aplikacije – vpišemo naziv in izberemo regijo (Slika 14).



Slika 14: Ustvarjanje aplikacije na platformi AppHarbor.

Ker spletna aplikacija za delovanje potrebuje podatkovno bazo ter nastavitve za pošiljanje pošte, moramo vključiti še nekatere dodatke. Vsaka nameščena aplikacija lahko vsebuje neomejeno število dodatkov, vendar so nekateri plačljivi. V katalogu dodatkov poiščemo dodatek SQL Server in Mailgun. Z brezplačno uporabo dobimo:

- podatkovni strežnik z 20 MB shrambe za podatkovno bazo,
- podatkovno bazo lahko upravljamo z zunanjimi orodji,
- možnost pošiljanja do 300 elektronskih sporočil dnevno,
- začasno shranjevanje prejete elektronske pošte.

Ob namestitvi dodatka se ustvarijo konfiguracijske spremenljivke, ki se naključno generirajo vsakokrat, ko namestimo dodatek. S pomočjo spremenljivk lahko dodatke uporabljamo v naši spletni aplikaciji oziroma v katerikoli drugi (na primer uporaba v namizni aplikaciji), zato si jih lahko predstavljamo, kot sistemske spremenljivke na lokalnem računalniku.

Da bo spletna aplikacija lahko uporabljala dodatke, jo moramo ustrezno nastaviti, za kar potrebujemo vrednosti spremenljivk, ki so shranjene v tabeli konfiguracijskih spremenljivk. Pred namestitvijo spletne aplikacije na platformo je pomembna predvsem spremenljivka

SQLSERVER_CONNECTION_STRING, ki vsebuje podatke za vzpostavljanje povezave s podatkovno bazo. Vrednost spremenljivke moramo vnesti v programsko kodo spletne aplikacije. Ostale spremenljivke lahko z aplikacijo povežemo potem, ko je ta že nameščena.

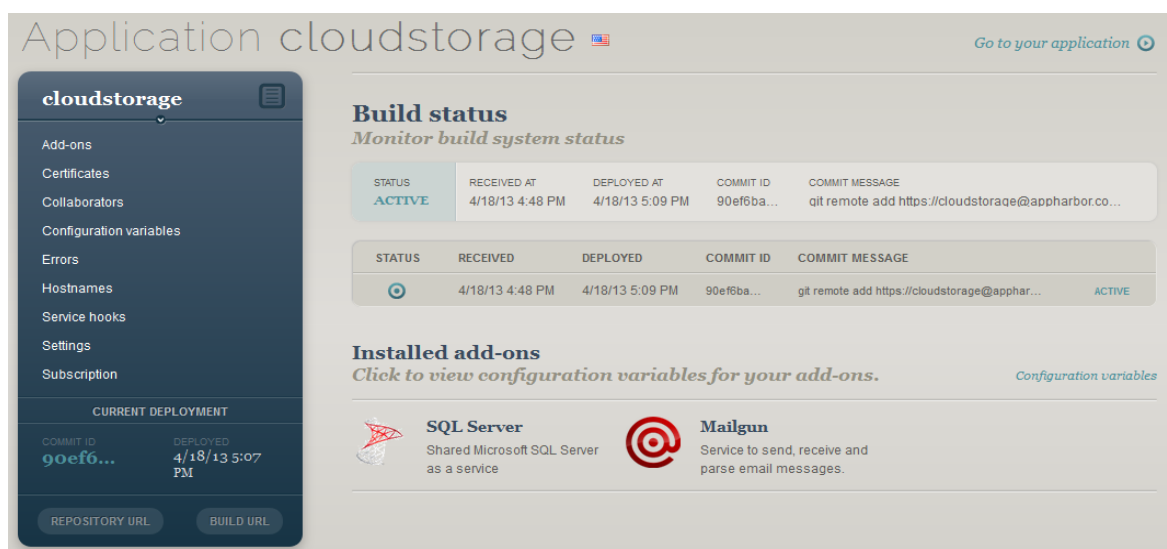
5.3.2 Namestitev na AppHarbor

V našem primeru smo platformo AppHarbor uporabili kot SQL strežnik, kot alternativno rešitev pa bomo opisali še postopek celotne namestitve.

Spletni uporabniški vmesnik ne omogoča prenosa izdelane spletne aplikacije na platformo. Za to moramo uporabiti dodatno programsko opremo, v našem primeru GitHub. Aplikacija zahteva avtentikacijo, zato je potrebna tudi registracija v storitvi GitHub.

V storitvi GitHub ustvarimo novo odlagališče oziroma repozitorij (angl. repository) ter na seznam kolaborantov (angl. collaborators) dodamo uporabnika »apphb«. S tem smo AppHarborju omogočili rabo naših datotek. Ustvarjen repozitorij moramo izbrati še v storitvi AppHarbor. Ustvari se t. i. most za prenos datotek med odjemalcem in strežnikom – storitvijo. Potrebna je še sinhronizacija, da se datoteke prenesejo na platformo. V program vnesemo naslov repozitorija in zaženemo sinhronizacijo.

Če programska koda nameščene aplikacije ne vsebuje napak, se na platformi prevede in ustvari aplikacija. Prav tako se ustvari URL naslov, preko katerega lahko odpremo našo aplikacijo. Slika 15 prikazuje uspešno nameščeno aplikacijo. V primeru napak se ustvari dnevnik z natančnim opisom napake.



The screenshot displays the AppHarbor application interface. On the left is a dark sidebar menu with options like 'Add-ons', 'Certificates', 'Collaborators', 'Configuration variables', 'Errors', 'Hostnames', 'Service hooks', 'Settings', and 'Subscription'. The main content area is titled 'Application cloudstorage' and includes a 'Build status' section with a table of build events. Below that is an 'Installed add-ons' section showing 'SQL Server' and 'Mailgun'.

STATUS	RECEIVED AT	DEPLOYED AT	COMMIT ID	COMMIT MESSAGE
ACTIVE	4/18/13 4:48 PM	4/18/13 5:09 PM	90ef6ba...	git remote add https://cloudstorage@appharbor.co...

STATUS	RECEIVED	DEPLOYED	COMMIT ID	COMMIT MESSAGE	
	4/18/13 4:48 PM	4/18/13 5:09 PM	90ef6ba...	git remote add https://cloudstorage@apphar...	ACTIVE

Icon	Add-on Name	Description
	SQL Server	Shared Microsoft SQL Server as a service
	Mailgun	Service to send, receive and parse email messages.

Slika 15: Aplikacija nameščena na AppHarbor.

Namestitev spletne aplikacije je s tem končana, vendar kot omenjeno ponudnik storitve AppHarbor ne podpira trajnega shranjevanja datotek. Aplikacija lahko shranjuje le začasne datoteke, ki se izbrišejo vsakokrat, ko namestimo novo verzijo aplikacije. Za trajno shranjevanje datotek uporabnikov moramo poskrbeti sami.

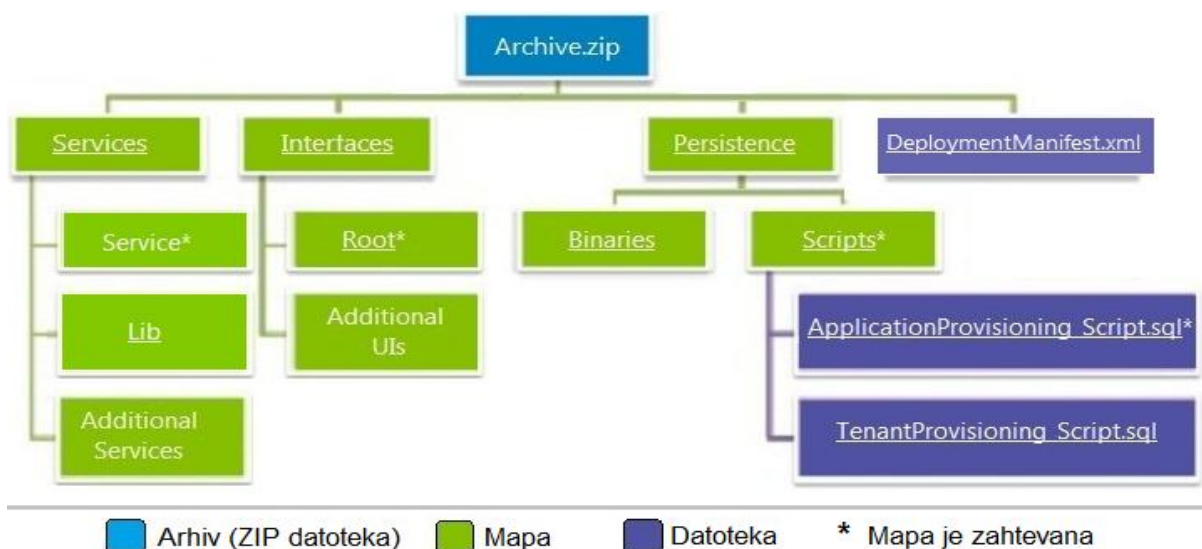
Kot rešitev za shranjevanje datotek ponudnik storitve ponuja shranjevanje na drugih platformah, ki podpirajo tehnologijo CORS. Datoteke bi tako lahko shranjevali v Amazon S3 ali Google Cloud. V tem primeru bi morali:

- Nadgraditi spletno aplikacijo, tako da ji dodamo programsko kodo za prenašanje datotek preko CORS.
- Spletni aplikaciji na platformi AppHarbor moramo dovoliti dostop do datotečnega sistema, tako da se ustvari začasna datoteka, ki se prenese k izbranemu ponudniku za shranjevanje.
- Konfigurirati uporabniški račun Amazon S3 ali Google Cloud za delo s CORS.

5.3.3 Namestitev na Apprendo

Kot pri ostalih storitvah se moramo tudi tukaj najprej registrirati, šele potem se lahko vpišemo in uporabljamo storitev. Čeprav je uporabnikom na voljo programska oprema za upravljanje nameščenih aplikacij, lahko namestitev opravimo tudi preko spletnega vmesnika.

Za namestitev aplikacij moramo upoštevati predpisano strukturo map in datotek projekta, kar prikazuje Slika 16.

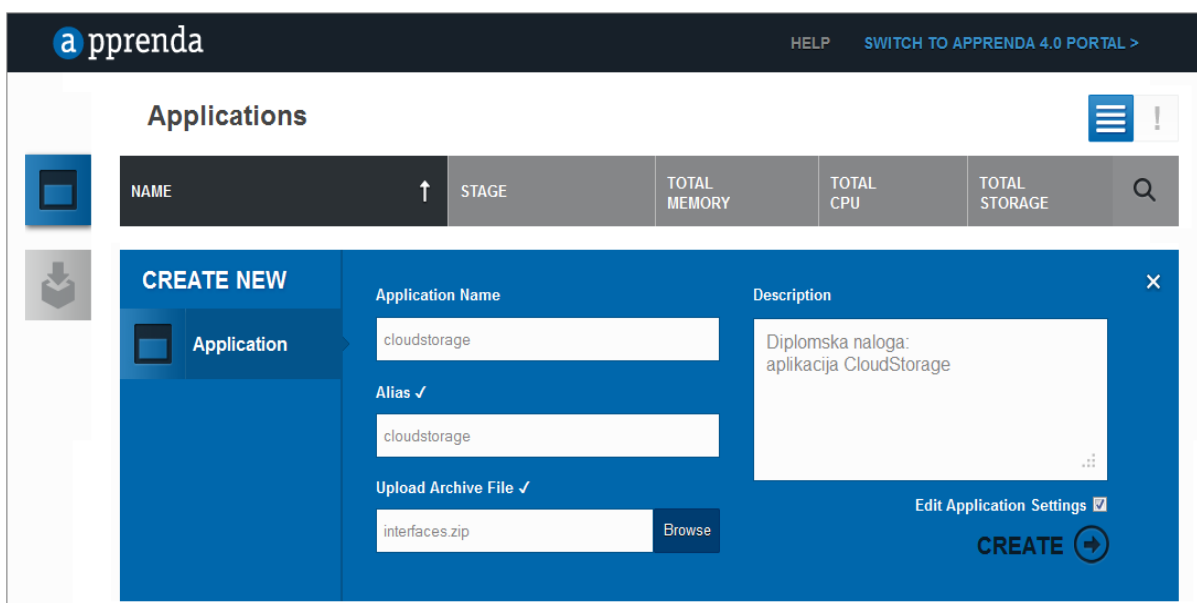


Slika 16: Struktura map in datotek projekta.

(vir: http://docs.apprenda.com/images/all/Archive_Structure.jpg)

Za namestitev naše aplikacije smo ustvarili mapo »interfaces«, ki lahko vsebuje datoteke za uporabniški vmesnik. Znotraj mape mora biti podmapa »root«, kamor smo shranili vse datoteke projekta.

Sedaj se lahko prijavimo v spletno storitev Apprenda in aplikacijo namestimo. Po prijavi smo preusmerjeni na seznam naših aplikacij. Tukaj si lahko ogledamo nastavitve naloženih aplikacij oziroma njihovo porabo virov. Za namestitev nove aplikacije izpolnimo obrazec, kar prikazuje Slika 17.



Slika 17: Nameščanje aplikacije na platformo Apprenda.

Namestitev, spreminjanje in uporaba aplikacije potekajo v t. i. življenjskem ciklu. Aplikacija se vedno nahaja v eni izmed stanj:

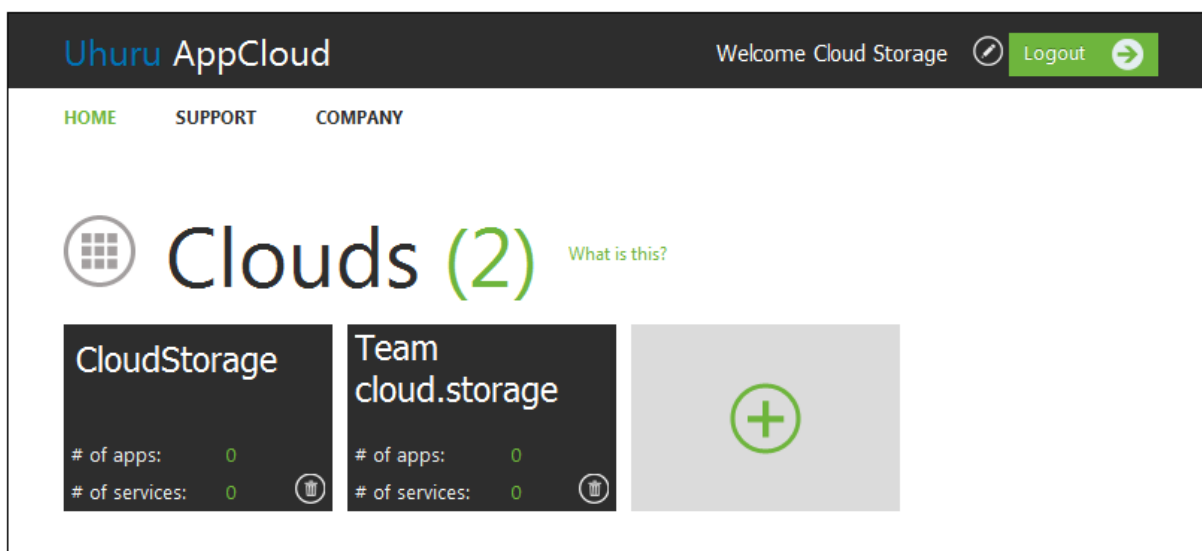
- Definicija (definition): Na začetku je aplikacija v stanju definicije. Lahko jo spreminjamo, dodajamo nove verzije, spreminjamo nastavitve in pravice aplikacije.
- Peskovnik (sandbox): To stanje je namenjeno testiranju aplikacij. Peskovnik je varnostni mehanizem, ki aplikacijam omejuje dostop do operacijskega sistema.
- Objavljeno (published): Ko je aplikacija v tem stanju, je pripravljena za uporabo in na voljo našim končnim uporabnikom.
- Arhivirano (archived): Medtem ko aplikacijo spreminjamo in testiramo, se vse starejše verzije shranijo, kar omogoča, da lahko izbiramo med verzijami.

Čeprav imamo lahko nameščenih več verzij aplikacije, ima vsaka nameščena aplikacija le eno objavljeno verzijo. Lahko pa dodamo verzije po tem, ko je aplikacija že objavljena.

Ko prenesemo prvo verzijo (ZIP datoteko), se aplikacija nahaja v stanju definicije. Za testiranje jo moramo nadgraditi v peskovnik, za končno uporabo pa jo je treba nadgraditi v stanje objavljeno (angl. published). S tem je postopek namestitve aplikacije končan.

5.3.4 Namestitev na Uhuru AppCloud

Za uporabo platforme Uhuru se lahko registriramo z enim od računov Google, Microsoft ali Facebook. Po registraciji se prijavimo z izbranim računom ter dovolimo ponudniku dostop do osnovnih podatkov in elektronskih naslovov, ki so povezani z računom. Preusmerjeni smo na seznam, kjer lahko pregledujemo in urejamo oblake. Ustvariti moramo nov oblak, kot prikazuje Slika 18.



Slika 18: Uporabniški vmesnik Uhuru AppCloud.

Tukaj naredimo nov oblak, v katerega bomo namestili spletno aplikacijo. Čeprav je na voljo programska oprema za nameščanje in upravljanje oblakov, smo za namestitev naše aplikacije uporabili program VMCU, ki se uporablja preko ukazne vrstice. Za uporabo programa smo morali na računalnik namestiti ogrodje Ruby in RubyGems. Za celoten postopek namestitve smo v ukazni vrstico vpisali ukaze:

```
vmcu target services.uhurucloud.com
```

```
vmcu login
```

```
vmcu cloud-team
```

```
vmcu push
```

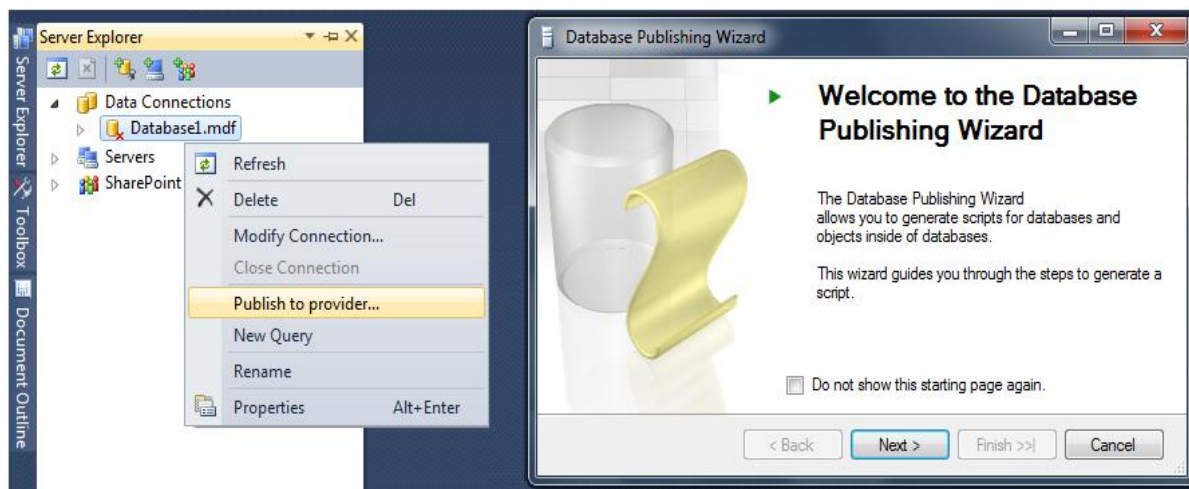
Najprej je treba vzpostaviti povezavo s storitvijo Uhuru in se nato vanjo prijaviti. Za prijavo se uporabljajo enkratna gesla. Vsakokrat, ko želimo v oblak namestiti aplikacijo, bo sistem generiral naključno geslo, ki ga uporabimo pri prijavi. Ukaz »cloud-team« nam izpiše oblake, ki smo jih ustvarili. Izbrati moramo enega od oblakov, v katerega želimo namestiti aplikacijo. Namestitev aplikacije začnemo z ukazom »push«, vpišemo pot do naše aplikacije in ji nastavimo konfiguracijo (spomin, dodatne storitve, število instanc).

5.4 Namestitev podatkovne baze

Po namestitvi spletne aplikacije na platformo smo morali poskrbeti še za podatkovno bazo. Med razvojem aplikacije smo podatkovno bazo ustvarili na lokalnem SQL strežniku, ki podatke shranjuje v MDF datoteko. Ob namestitvi aplikacije na platformo je treba prenesti še podatkovno bazo, ki se mora nahajati na oddaljenem SQL strežniku, na katerega se povezuje spletna aplikacija.

Podatkovne baze ni treba ponovno izdelati, saj lahko izdelamo njeno kopijo. Celotno bazo in vnesene podatke lahko shranimo v SQL skripto. To je datoteka z naborom SQL stavkov, ki določajo shemo podatkovne baze (tabele, atributi, podatki). Na ta način lahko kadarkoli ustvarimo podatkovno bazo na kateremkoli drugem podatkovnem strežniku, ne da bi morali poznati njeno zgradbo.

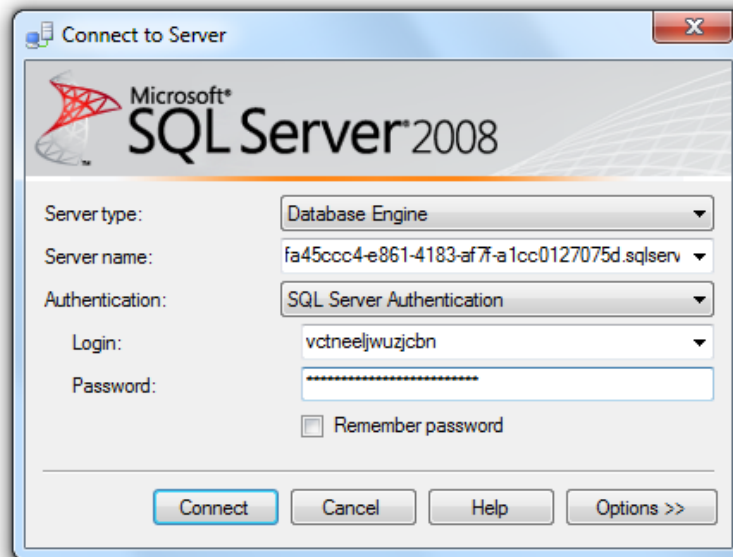
V razvojnem okolju Visual Studio poiščemo podatkovno bazo in v izbirnem meniju izberemo možnost »Publish to provider ...«. Sledimo dialogu (Slika 19), kjer lahko nastavimo dodatne možnosti in poimenujemo končno datoteko. Po končanem postopku se ustvari omenjena SQL datoteka.



Slika 19: Izdelava SQL skripte z orodjem Visual Studio.

V drugem koraku moramo iz nastale datoteke ustvariti podatkovno bazo. Za ta namen smo uporabili Microsoftovo orodje SQL Server Management Studio 2008. Ko programsko opremo zaženemo, moramo opraviti prijavo na oddaljeni strežnik.

Podatke za prijavo dobimo pri ponudniku storitve SQL strežnik. V našem primeru je to AppHarbor. Iz spremenljivke `SQLSERVER_CONNECTION_STRING` moramo ločiti 3 podatke, s katerimi se lahko prijavimo: naslov strežnika, uporabniško ime, geslo.



Slika 20: Vzpostavitev povezave z oddaljenim SQL strežnikom.

Po uspešni prijavi v SQL Server Management Studio odpremo skripto in jo zaženemo. Po končanem postopku se ustvari podatkovna baza. Podatki, ki so ostali shranjeni v lokalni podatkovni bazi, se shranijo tudi na oddaljeni SQL strežnik.

5.5 Nastavitev spletne aplikacije

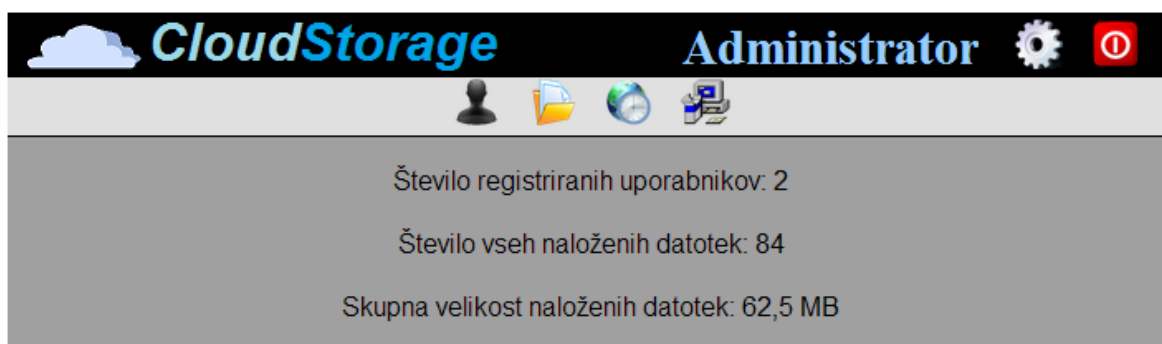
Med razvojem spletne aplikacije so se pojavila naslednja vprašanja: Kam shranjevati datoteke? Kolikšno kapaciteto bodo imeli uporabniki? Kateri elektronski naslov uporabiti za pošiljanje pošte uporabnikom? Vse te nastavitve je mogoče definirati neposredno v programski kodi, vendar tak pristop ni primeren, ker:

- moramo že ob razvoju poznati omejitve s strani ponudnika platforme,
- bi vsaka sprememba zahtevala ponovno namestitev aplikacije v oblak.

Rešitev smo našli v razširitvi podatkovne baze z dodatno tabelo, kjer so shranjene vse nastavitve. Ker je podatkovna baza ločena od same aplikacije, lahko podatke kadarkoli spremenimo. Prednost pa je tudi univerzalni dostop do nastavitvev, saj potrebujemo le prijavnne podatke, da se povežemo s SQL strežnikom. Zaradi tega smo spletno aplikacijo vgradili možnost administracije, ki je na voljo preko spletnega vmesnika. Administracija je možna iz kateregakoli računalnika, saj je potreben le dostop do svetovnega spleta. Osnovni namen administracije je:

- nastavitve privzete kapacitete uporabnikom za shranjevanje,
- določanje mape v datotečnem sistemu, kamor se shranjujejo datoteke uporabnikov,
- upravljanje SMTP nastavitvev za pošiljanje elektronske pošte,
- določanje podprtih tipov datotek: ikona, opis, MIME podatki,
- nadzor nad uporabniškimi računi: brisanje neaktiviranih računov.

Preden spletno aplikacijo začnejo uporabljati končni uporabniki, moramo nastaviti še nekatere nujne nastavitve. V brskalnik vpišemo URL naslov, kjer je nameščena aplikacija, in se prijavimo kot administrator. Po uspešni prijavi smo preusmerjeni na začetno stran (Slika 21), kjer so prikazani podatki o rabi spletne aplikacije. Uporabniški vmesnik vsebuje gumba za upravljanje tabel podatkovne baze: uporabniški računi, podprti tipi datotek, časovni pasovi, splošne nastavitve.



Slika 21: Začetna stran s prikazom informacij o rabi storitve.

V meniju kliknemo gumb za dostop do splošnih nastavitvev aplikacije (Slika 22) in v obrazec vpišemo zahtevane podatke:

- URL naslov, na katerem se nahaja spletna aplikacija,
- SMTP podatki in elektronski naslov za pošiljanje elektronske pošte,
- privzeta kapaciteta uporabniških računov in mapa za shranjevanje datotek.

1. Podatki o računu in storitvi:	
Uporabniško ime:	Administrator
Email - administrator:	cloudstorage@gmail.com
Geslo - administrator:
URL naslov:	http://apps.my.appendacloud.com
Zadnja sprememba:	25.5.2013 18:38:12
2. Elektronska pošta:	
Elektronski naslov:	postmaster@app5828.mailgun.org
Uporabniško ime:	postmaster@app5828.mailgun.org
Geslo:
SMTP strežnik:	smtp.mailgun.org
SMTP vrata:	587
3. Nastavitve za shranjevanje:	
Zasedenost diska:	65536718 B (62,5 MB)
Mapa za shranjevanje:	C:\Uporabniki
Kapaciteta računov:	100 MB
<input type="button" value="Preizkusi e-pošto"/> <input type="button" value="Shrani"/> <input type="button" value="Prekliči"/>	

Slika 22: Splošne nastavitve aplikacije.

URL naslov mora biti pravilen, saj predstavlja absolutno pot do spletnih obrazcev, v primeru, ko je uporabniku poslana povezava (potrditev registracije uporabniškega računa ali obnovitev računa). Če bi administrator vpisal napačen URL naslov, bi uporabnik prejel napačno povezavo in tako ne bi mogel opraviti zahtevanega dejanja.

Podatki o elektronski pošti morajo biti pravilni. V nasprotnem primeru uporabnikom ne bo mogoče poslati potrditvene pošte in posledično tudi prijava v aplikacijo ne bo možna. Administratorju je zato na voljo pošiljanje preizkusne pošte. Pri nastavljanju elektronske pošte smo uporabili naslov, ki smo ga pridobili od ponudnika AppHarbor, lahko pa bi uporabili katerikoli drugi naslov, na primer Gmail, Hotmail itd. Paziti moramo le, da glede na uporabljen naslov pravilno nastavimo SMTP strežnik ter vrata.

Preden določimo kapaciteto uporabniških računov, se je potrebno prepričati, kolikšno kapaciteto imamo sploh na voljo. Če bi vpisali preveč, bi imeli uporabniki napačne informacije o zasedenosti. Po shranitvi podatkov je aplikacija pripravljena za uporabo za končne uporabnike.

5.6 Vzdrževanje spletne aplikacije

Po tem, ko je aplikacija že nameščena na platformo, lahko administrator spreminja nekatere nastavitve. Preko vgrajenih spletnih obrazcev lahko pregleduje in ureja podatke v nekaterih tabelah podatkovne baze.

Časovni pasovi

Za pravilen prikaz datuma in časa morajo uporabniki imeti nastavljen ustrezen časovni pas. Podatkovna baza vsebuje tabelo časovnih pasov, kjer so shranjene posamezne nastavitve. Administrator lahko doda različne regije in določi časovne odmike le-teh. Pri shranjevanju časovnih vrednosti se vedno shrani čas, ki je nastavljen na strežniku – platformi. Ob prikazu se upošteva še shranjena nastavitvev.

Uporabniški računi

Zaradi varovanja osebnih podatkov se osebni podatki uporabnika administratorju ne pokažejo. Prikazani so le podatki o uporabniškem računu: uporabniško ime, datum kreiranja, kapaciteta in stanje računa. Administrator lahko spremeni kapaciteto ali izbriše uporabniške račune, ki so starejši od 7 dni in v tem času niso bili potrjeni.

Registrirani tipi datotek

Za predstavitev datotek in map potrebujemo informacije o tipu le-teh. Na ta način lahko za vsako datoteko določimo:

- ikono datoteke: s tem določimo, katera ikona bo prikazana uporabnikom;
- vrsto (kategorijo): tako je uporabnikom omogočeno lažje iskanje datotek, npr. slike, glasba, besedilo, video itd.;
- opis datoteke: datoteko lahko bolj podrobno opišemo, kar bodo uporabniki videli v podrobnostih o datoteki;
- MIME podatke: za prenos datoteke potrebujemo podatke, ki povedo, kakšno vrsto datoteke prenašamo.

V podatkovno bazo lahko administrator doda različne tipe datotek. Upoštevati moramo tudi možnost, da ima uporabnik lahko shranjene tudi druge formate datotek, ki niso shranjeni v podatkovni bazi. V tem primeru se uporabijo privzete nastavitve, ki so definirane v programu.

6 Zaključek

Področje računalništva v oblaku se hitro razvija; nastajajo nove storitve, ki prinašajo nove funkcionalnosti. Kar je nekoč uporabnike le dodatno navduševalo, je danes nujno potrebno. Sklepamo lahko, da se bo v prihodnosti računalništvo v oblaku še naprej širilo in postalo še bolj priljubljeno med uporabniki in podjetji.

Diplomsko delo se osredotoča predvsem na postopek razvoja in namestitve programske opreme v oblak. Za dosežen rezultat je potrebno znanje s področja računalništva v oblaku, zato smo se v uvodnih poglavjih posvetili predvsem teoretičnim osnovam oblakov. Poiskali smo tudi nekaj aktualnih storitev za shranjevanje datotek in preučili njihove funkcionalnosti. Na podlagi pridobljenih informacij smo določili zahteve in izdelali lastno aplikacijo.

Spletno aplikacijo je sicer mogoče izdelati še v mnogih drugih programskih jezikih, a je vseeno priporočljivo prej preveriti ponudnike platform, saj se lahko njihove ponudbe zelo razlikujejo, tako v podpori kot tudi ceni. Za namestitev naše aplikacije smo iskali možnosti za brezplačno postavitve, vendar je treba poudariti, da so sredstva ponudnikov zelo omejena. Za naše potrebe so sredstva, ki so na voljo (kapaciteta, procesorska moč itd.), zadostovala, toda smiselno bi jih bilo nadgraditi, če bi storitev uporabljalo veliko uporabnikov.

Med samim razvojem aplikacije smo se osredotočili predvsem na funkcionalnosti storitve in njihovo izvedbo. Poskrbeti pa je treba tudi za varnost podatkov in pravilno delovanje aplikacije, zato je treba upoštevati vse možne situacije. Ravno zato je treba dati velik poudarek načrtovanju, saj dober načrt zmanjša možnosti težav pri razvoju. Po končanem razvoju pa ne smemo pozabiti na testiranje, saj so prav uporabniki tisti, ki odločajo, ali bo programska oprema uspešna. Čeprav aplikacija izpolnjuje svoj namen, obstaja še veliko možnosti za nadgradnjo. Nekaj smo jih opisali, a uporabniki bi še vedno dobili nove ideje.

7 Priloge

Priloga 1: Diagrami delovanja aplikacije

V nadaljevanju je opisanih nekaj primerov uporabe spletne aplikacije. Pri opisovanju dogodkov (razen registracije uporabnika) bomo upoštevali, da so izpolnjeni predpogoji za uporabo funkcionalnosti:

- uporabnik je prijavljen v spletno aplikacijo,
- parametri v URL naslovu dokumenta so pravilni in jih uporabnik ni spremenil.

Lahko bi se zgodilo, da se uporabnik odjavi iz aplikacije, nato pa preklopi na drugi zavihek/okno, kjer je obrazec še vedno prikazan. Če tudi bi kliknil na neki gumb, se operacija ne bi izvedla, saj se ob nalaganju spletnega obrazca preveri prijavljenost uporabnika. Ker se je uporabnik že prej odjavil, bo preusmerjen na prijavno stran.

Problem se pojavi tudi ob klicu spletnega dokumenta. Sistem deluje tako, da se podatki med obrazci prenašajo preko parametrov v URL naslovu. Uporabnik bi lahko vpisane parametre spremenil, kar bi lahko predstavljalo neko drugo datoteko (mogoče drugega uporabnika) ali pa bi bili podatki napačni. Zaradi tega se med nalaganjem dokumenta preveri veljavnost podatkov. Če so podatki nepravilni oziroma če uporabnik nima pravice dostopa do datoteke, se ne bo nič zgodilo.

Upoštevat moramo tudi, da v nekaterih primerih določene operacije niso možne, čeprav je gumb prikazan. Tako na primer uporabnik v mapi Koš ne more ustvarjati novih datotek, saj ta vsebuje le izbrisane datoteke. V tem primeru je treba uporabnika opozoriti. Za prikaz opozorila smo uporabili JavaScript funkcijo in jo povezali z ustreznim gumbom.

Kopiranje mape

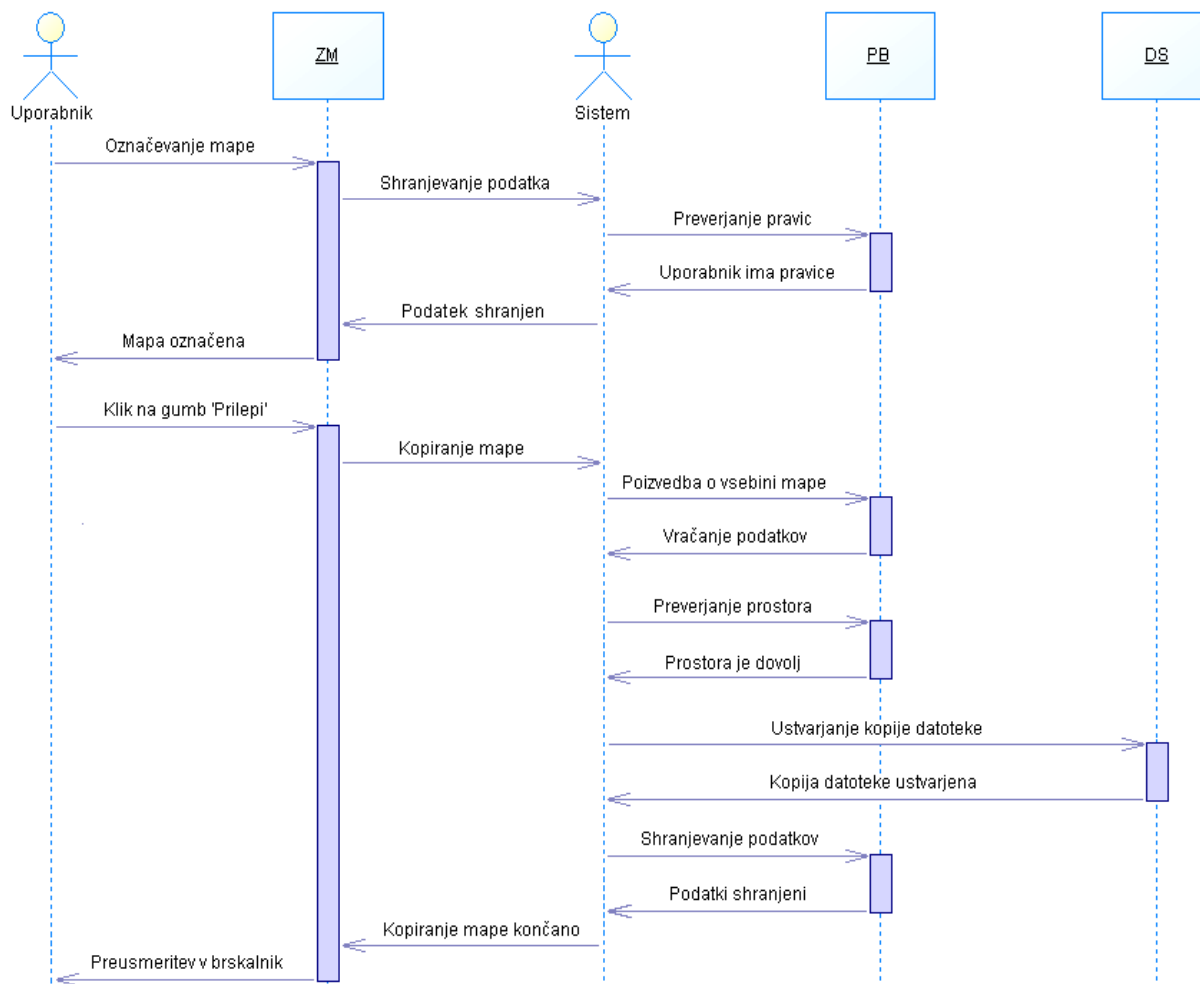
Opis uporabe:

Uporabnik želi kopirati mapo na neko drugo mesto. Operacija kopiranja poteka na principu kopiraj–prilepi (angl. copy-paste), zato mora uporabnik najprej določiti želeno mapo, jo odpreti ter nato vanjo »prilepiti« določeno mapo.

Osnovni tok dogodkov:

1. Uporabnik s klikom na gumb določi mapo za kopiranje.
2. Uporabnik poišče mesto (mapo), kamor želi premakniti določeno mapo.
3. Uporabnik s klikom na gumb 'Prilepi' začne operacijo premikanja.
4. Sistem preveri obstoj označene mape in pridobi njeno vsebino.

5. Sistem naredi ustrezno strukturo map (podmape) in podatke shrani v podatkovno bazo.
6. Sistem izvede zaporedje:
 - 6.1 Preveri, ali je kopiranje mogoče (glede na velikost datoteke).
 - 6.2 Datoteka se kopira na določeno mesto (mapo) na disku.
 - 6.3 Podatki o novonastali datoteki se shranijo v podatkovno bazo.
7. Operacija kopiranja je končana. Uporabniku se prikaže vsebina odprte mape.



Slika 23: Sekvenčni diagram za primer kopiranja mape.

Možne alternative:

1. Uporabnik je določeno mapo izbrisal, še preden je sprožil kopiranje. Postopek kopiranja v tem primeru ni mogoč. Uporabniku se prikaže vsebina odprte mape.
2. Uporabnik nima pravice za kopiranje v izbrano mesto. Postopek kopiranja je prekinjen.
3. Med kopiranjem je porabljena vsa kapaciteta. Postopek kopiranja se predčasno zaključi.
4. Na izbranem mestu že obstaja mapa z enakim imenom (in končnico). Mapa, ki jo kopiramo, je hkrati tudi preimenovana.

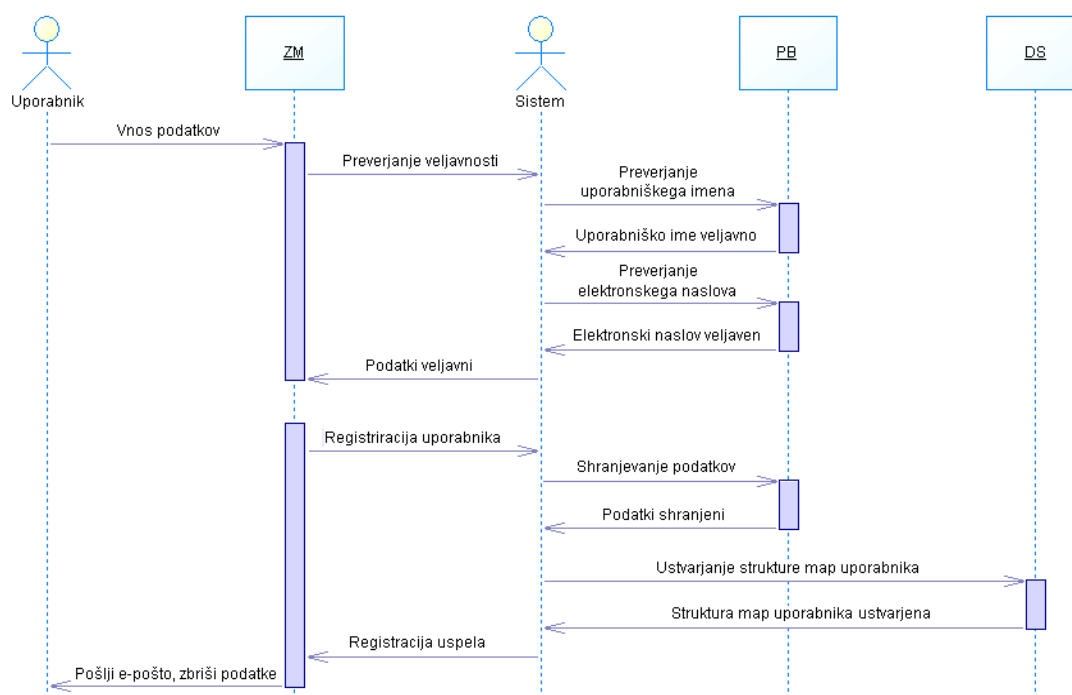
Registracija uporabnika

Opis uporabe:

Uporabnik si želi ustvariti uporabniški račun, s katerim se bo lahko prijavil v aplikacijo in uporabljal njene funkcije. V obrazcu za registracijo navesti svoje podatke. Če so podatki veljavni, bo po elektronski pošti prejel povezavo za potrditev računa.

Osnovni tok dogodkov:

1. Uporabnik na prijavni strani izbere možnost 'Ustvari mi nov račun'.
2. Sistem uporabnika preusmeri na stran za registracijo.
3. Uporabnik mora vpisati svoje podatke in jih potrditi s klikom na gumb 'Potrdi'.
4. Sistem preveri veljavnost vpisanih podatkov.
5. Podatki o uporabniškem računu se shranijo v podatkovno bazo. Na disku se kreira struktura map za shranjevanje datotek uporabnika.
6. Sistem pošlje potrditveno pošto na naveden uporabniški račun. Vpisani podatki se izbrišejo.
7. Uporabnik mora odpreti elektronsko pošto in potrditi registracijo s klikom na povezavo.



Slika 24: Sekvenčni diagram za primer registracije uporabnika.

Možne alternative:

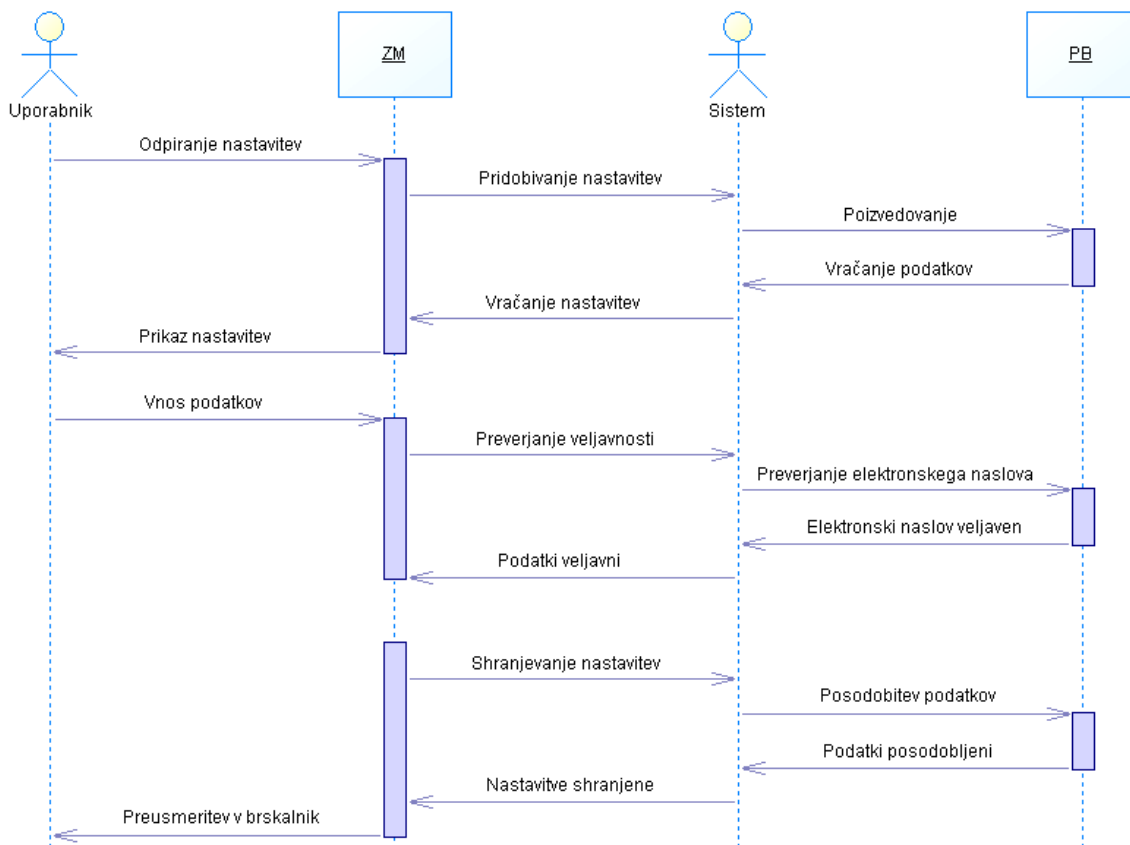
1. Uporabnik je preklical registracijo. Sistem preusmeri uporabnika na prijavno stran.
2. Vsaj en od vpisanih podatkov ni veljaven: vsebuje premalo znakov, preveč znakov ali pa neveljavne znake.
3. Uporabnik nima dostopa do elektronskega naslova, zato ne more potrditi registracije: uporabnik se je lahko zmotil pri vpisu podatka.

Spreminjanje nastavitve računa

Uporabnik želi spremeniti nastavitve svojega uporabniškega računa. Za ogled in spreminjanje podatkov mora klikniti na gumb 'Nastavitve'. Odpre se obrazec, kjer so prikazani podatki o računu. Uporabnik spremeni podatke in jih potrdi s klikom na gumb 'Shrani'.

Osnovni tok dogodkov:

1. Uporabnik s klikom na gumb odpre nastavitve.
2. Sistem pridobi podatke o uporabniškem računu in jih prikaže v obrazcu.
3. Uporabnik spremeni podatke in uveljavi spremembe s klikom na gumb 'Shrani'.
4. Sistem preveri veljavnost vpisanih podatkov.
5. Podatki o uporabniškem računu se shranijo v podatkovno bazo.
6. Obrazec se zapre. Uporabnik je preusmerjen v brskalnik datotek.



Slika 25: Sekvenčni diagram za primer spreminjanja nastavitve.

Možne alternative:

1. Vsaj en podatek ni veljaven. Sistem uporabi prejšnji podatek in shrani nastavitve.
2. Uporabnik prekliče shranjevanje s klikom na gumb 'Prekliči'. Nastavitve niso shranjene.
3. Uporabnik klikne drugi gumb (na primer Deaktiviraj račun). Nastavitve niso shranjene, ker se izvede druga operacija. Po potrebi je uporabnik preusmerjen drugam.

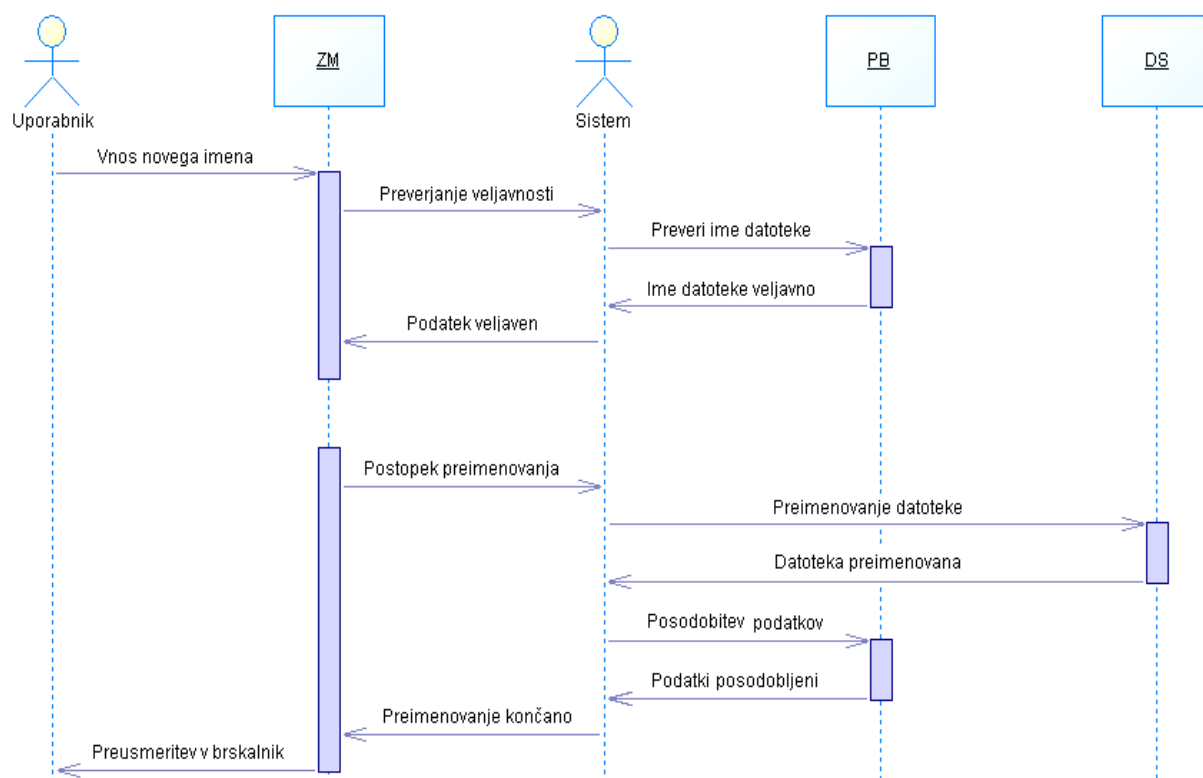
Preimenovanje datoteke

Opis uporabe:

Uporabnik želi preimenovati datoteko. Za vpis novega imena mora klikniti ustrezní gumb poleg imena datoteka, s čimer odpre obrazec. Vpisati mora novo ime in ga potrditi.

Osnovni tok dogodkov:

1. Uporabnik s klikom na gumb odpre obrazec za preimenovanje datoteke.
2. Uporabnik vpiše novo ime datoteke in s klikom na gumb 'Potrdi' uveljavi spremembo.
3. Sistem preveri veljavnost imena datoteke.
4. Datoteka na disku se preimenuje.
5. Podatek o imenu datoteke v podatkovni bazi se posodobi.
6. Sistem zapre obrazec in preusmeri uporabnika na v brskalnik datotek.



Slika 26: Sekvenčni diagram za primer preimenovanja datoteke.

Možne alternative:

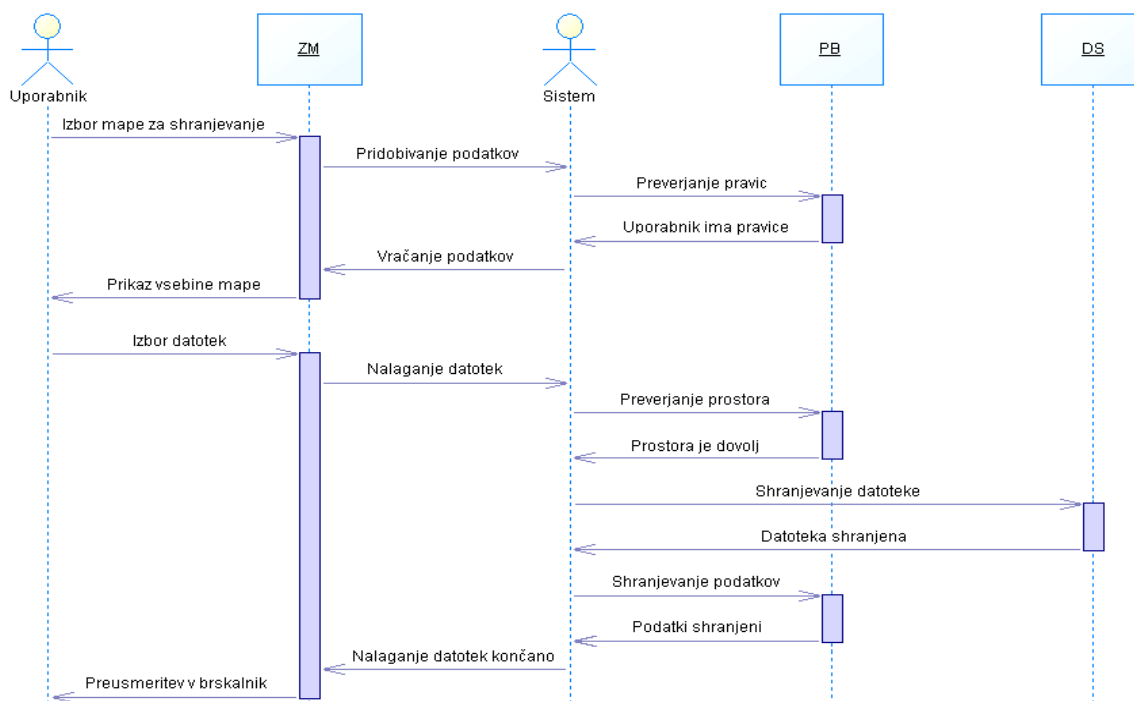
1. Uporabnik zapre obrazec s klikom na gumb 'Prekliči'. Sledi preusmeritev na izbrano mesto.
2. Uporabnik je v novem imenu datoteke uporabil nedovoljene znake ali pa novo ime vsebuje preveč znakov.
3. Datoteka z enakim imenom in končnico že obstaja. Izbrana datoteka ni preimenovana. Obrazec se zapre in uporabnik je preusmerjen v brskalnik datotek.

Nalaganje datotek z računalnika

Uporabnik želi naložiti datoteke s svojega računalnika. Odpre mora obrazec za nalaganje, kjer lahko izbere do 10 datotek.

Osnovni tok dogodkov:

1. Uporabnik poišče mesto (mapo), kamor želi naložiti datoteke.
2. Uporabnik s klikom na gumb odpre obrazec za nalaganje datotek.
3. Uporabnik izbere datoteke, ki jih želi naložiti in klikne gumb 'V redu'.
4. Sistem pridobi vsebino mape, kamor se bodo shranile datoteke.
5. Sistem izvede zaporedje:
 - 5.1 Preveri, ali je nalaganje mogoče (glede na velikost datoteke).
 - 5.2 Shrani datoteko na izbrano mesto na disku.
 - 5.3 Podatki o naloženi datoteki se shranijo v podatkovno bazo.
6. Postopek nalaganja je končan. Obrazec za nalaganje se zapre in uporabnik je preusmerjen v mapo, kjer so shranjene novonastale datoteke.



Slika 27: Sekvenčni diagram za primer nalaganja datoteke.

Možne alternative:

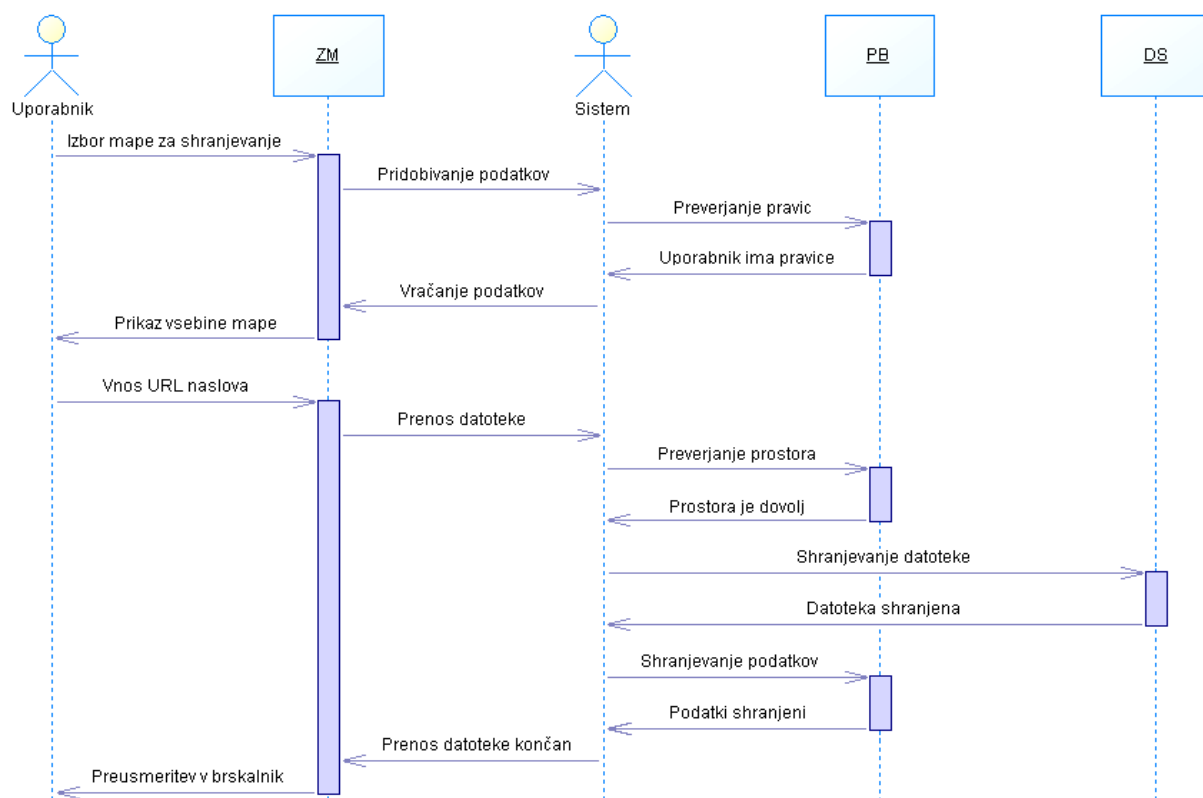
1. Uporabnik nima pravice za nalaganje datotek na izbrano mesto.
2. Uporabnik je preklical nalaganje s klikom na gumb 'Prekliči'.
3. Med nalaganjem je porabljena vsa kapaciteta. Postopek nalaganja se predčasno zaključi.
4. Na izbranem mestu že obstaja datoteka z enakim imenom in končnico. Naložena datoteka je hkrati tudi preimenovana.

Prenos datoteke s spleta

Uporabnik želi prenesti datoteko neposredno s spleta. Postopek prenašanja poteka tako, da uporabnik najprej odpre želeno mapo, nato odpre obrazec, kamor vpiše URL naslov.

Osnovni tok dogodkov:

1. Uporabnik poišče mesto (mapo), kamor želi naložiti datoteko.
2. Uporabnik s klikom na gumb odpre obrazec za prenos datoteke s spleta.
3. Uporabnik vpiše URL naslov datoteke in ga potrdi s klikom na gumb 'V redu'.
4. Sistem pridobi vsebino mape, kamor se bo datoteka shranila.
5. Sistem prenese datoteko s spleta in jo shrani na izbrano mesto na disku.
6. Sistem pridobi podatke o datoteki in jih shrani v podatkovno bazo.
7. Prenos datoteke je končan. Obrazec za nalaganje se zapre in uporabnik je preusmerjen v mapo, kjer je shranjena novonastala datoteka.



Slika 28: Sekvenčni diagram za primer prenosa datoteke s spleta.

Možne alternative:

1. Uporabnik nima pravice za shranjevanje na izbrano mesto, zato se obrazec ne prikaže.
2. Uporabnik prekliče prenos s klikom na gumb 'Prekliči'.
3. Vpisani URL naslov je neveljaven. Datoteka se ne prenese.
4. Na izbranem mestu že obstaja datoteka z enakim imenom in končnico. Prenesena datoteka se hkrati tudi preimenuje
5. Med prenosom je bila porabljena kapaciteta. Datoteka se zbríše, spremembe se razveljavijo.

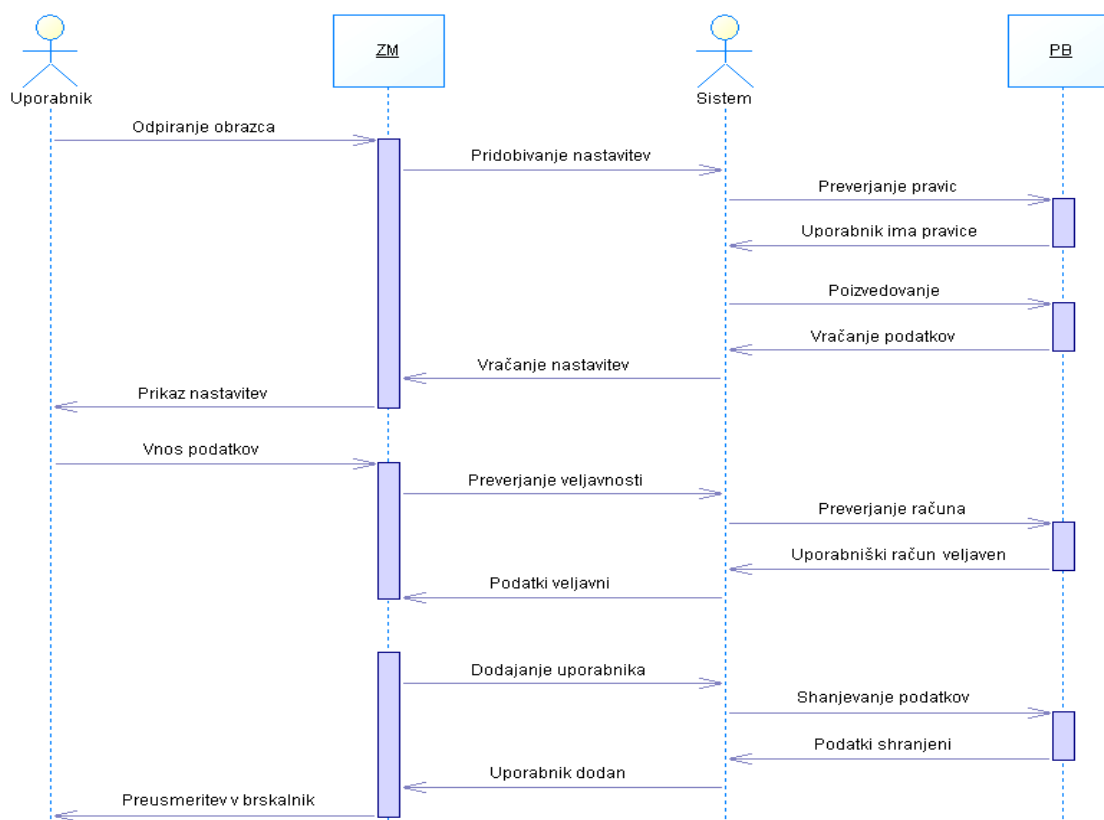
Deljenje datoteke z drugim uporabnikom

Opis uporabe:

Uporabnik želi datoteko deliti z drugimi uporabniki. Poleg imena datoteke mora klikniti na ustrezen gumb, s čimer odpre obrazec za skupno rabo. Vpisati mora uporabniško ime in ga s klikom na gumb 'Dodaj' dodati na seznam.

Osnovni tok dogodkov:

1. Uporabnik s klikom na gumb odpre obrazec za urejanje skupne rabe.
2. Sistem pridobi podatke o skupni rabi datoteke in jih prikaže v obrazcu.
3. Uporabnik vpiše uporabniško ime in ga potrdi s klikom na gumb 'Dodaj'.
4. Sistem poišče navedenega uporabnika in ga prikaže na seznamu.
5. Uporabnik zapre obrazec klikom na gumb 'V redu'.



Slika 29: Sekvenčni diagram za primer deljenja datotek z drugimi uporabniki.

Možne alternative:

1. Uporabnik prekliče deljenje datoteke. Sistem ga preusmeri v brskalnik datotek.
2. Navedeno uporabniško ime ne obstaja. Uporabnik ni dodan, spremembe niso shranjene.
3. Naveden uporabniški račun ni aktiviran ali pa je blokiran. Uporabnik ni dodan, spremembe niso shranjene.
4. Uporabnik omogoči prenos datotek komurkoli v spletu. Četudi je uporabniško ime vpisano, sistem upošteva alternativno možnost.

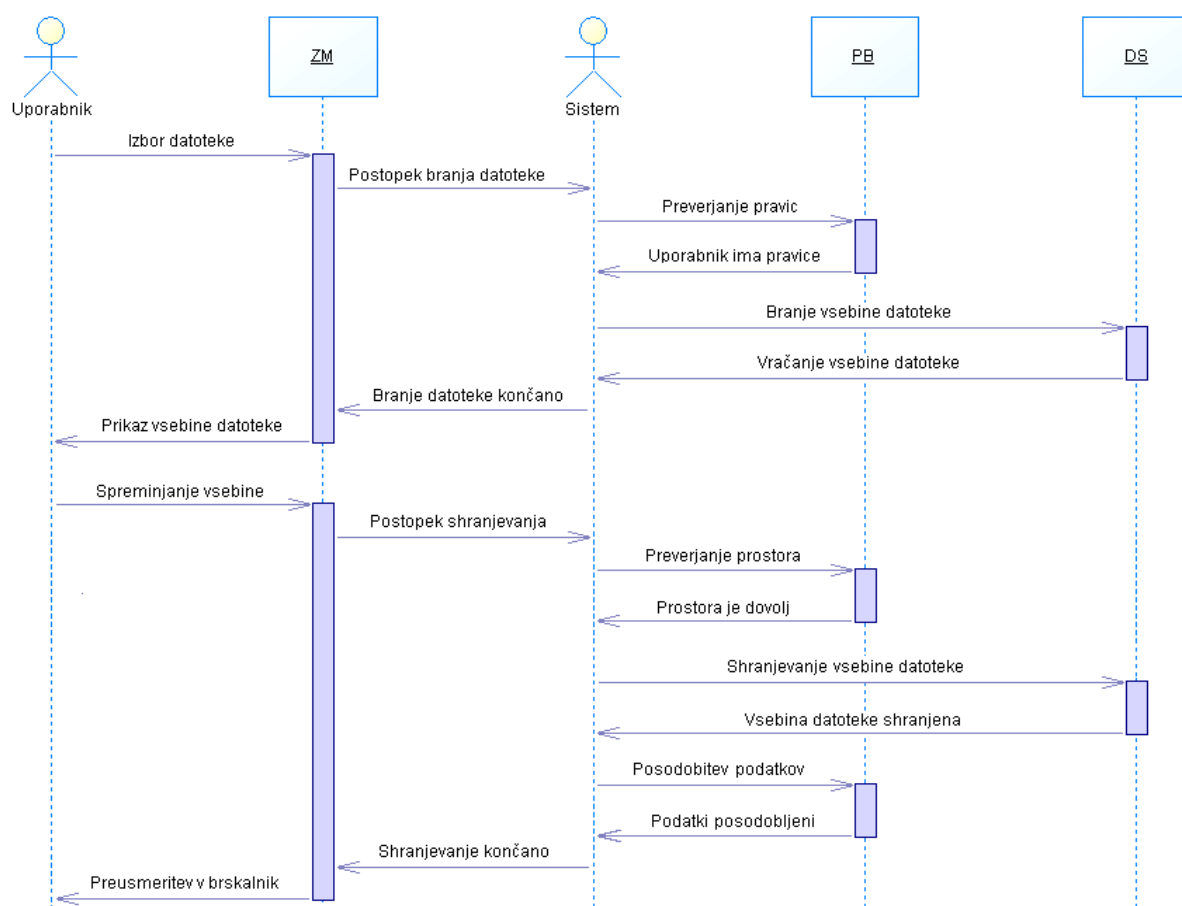
Urejanje vsebine datoteke

Opis uporabe:

Uporabnik želi urediti vsebino datoteke. Poleg imena datoteke mora klikniti na ustrezen gumb. Preusmerjen je v urejevalnik, kjer se izpiše vsebina datoteke. Uporabnik spremeni vsebino in datoteko shrani.

Osnovni tok dogodkov:

1. Uporabnik s klikom na gumb odpre urejevalnik datoteke.
2. Sistem prebere datoteko in preveri, ali ne vsebuje posebnih znakov.
3. Sistem prikaže vsebino datoteke.
4. Uporabnik uredi datoteko in shrani novo vsebino s klikom na gumb 'Shrani'.
5. Sistem preveri novo velikost datoteke, datoteka se shrani, podatki v PB se posodobijo.
6. Uporabnik je preusmerjen na mesto.



Slika 30: Sekvenčni diagram za primer urejanja vsebine datoteke.

Možne alternative:

1. Datoteka vsebuje posebne znake. Vsebina ni prikazana
2. Uporabnik priključuje shranjevanje s klikom na gumb 'Prekliči'.
3. Porabljena je vsa kapaciteta. Datoteke ni mogoče shraniti.

8 Viri in literatura

- [1] Ben Halpert: *Auditing Cloud Computing: A security and privacy guide*, Indianapolis: Cisco Press, 2012 (strani 4 – 9)
- [2] Josyula Venkata, Orr Malcolm, Page Greg: *Cloud Computing: Automating the virtualized data center*, Hoboken : J. Wiley & Sons, 2011 (strani: 11 – 14)
- [3] ComputerWeekly – *A history of cloud computing* (2009). Dostopno 30.3.2013 na: <http://www.computerweekly.com/feature/A-history-of-cloud-computing>
- [4] Toolbox – *Why is it called 'Cloud computing'* (2009). Dostopno 30.3.2013 na: <http://it.toolbox.com/blogs/original-thinking/why-is-it-called-quotcloud-computingquot-30713>
- [5] Mladi podjetnik – članek *Računalništvo v oblaku* (2011). Dostopno 3.4.2013 na: <http://mladipodjetnik.si/podjetniski-koticek/poslovanje/racunalnistvo-v-oblaku-fleksibilnejsi-dostop-do-racunalniskih-storitev>
- [6] Cloud Computing: *Is it really all that beneficial* (2013). Dostopno 8.4.2013 na: <http://mobiledevices.about.com/od/additionalresources/a/Cloud-Computing-Is-It-Really-All-That-Beneficial.htm>
- [7] Blog – *Advantages, disadvantages, benefits and risks* (2012). Dostopno 8.4.2013 na: <http://theprofessionalspoint.blogspot.com/2012/11/advantages-disadvantages-benefits-and.html>
- [8] Spletna enciklopedija Wikipedia – iskalni niz: cloud computing. Dostopno 14.4.2013 na: http://en.wikipedia.org/wiki/Cloud_computing
- [9] SecurityWeek – *Categories of cloud computing* (2011). Dostopno 15.4.2013 na: <http://www.securityweek.com/three-categories-cloud-computing-whats-your-flavor>
- [10] Spletna enciklopedija Wikipedia – iskalni niz: »dropbox«. Dostopno 23.4.2013 na: http://en.wikipedia.org/wiki/Dropbox_%28service%29
- [11] Spletna stran storitve Dropbox. Dostopno 23.4.2013 na: <https://www.dropbox.com/pricing>
- [12] Spletno mesto Techcrunch (2013). Dostopno 29.4.2013 na: <http://techcrunch.com/2013/02/05/15-million-users-strong-box-debuts-new-partner-network-to-expand-cloud-and-collaboration-service/>
- [13] Spletna stran storitve Box. Dostopno 29.4.2013 na: <https://www.box.com/pricing/>
- [14] Spletna enciklopedija Wikipedia – iskalni niz: »google drive«. Dostopno 30.4.2013 na: http://en.wikipedia.org/wiki/Google_Drive

- [15] Spletno mesto Liveside – članek *Skydrive* (2013). Dostopno 3.5.2013 na:
<http://www.liveside.net/2013/05/06/skydrive-reaches-250-million-users-a-significant-number/>
- [16] Spletna enciklopedija Wikipedia – iskalni niz: »skydrive«. Dostopno 3.5.2013 na:
<http://en.wikipedia.org/wiki/SkyDrive>
- [17] Spletna enciklopedija Wikipedia – iskalni niz: »ubuntu one«. Dostopno 6.5.2013 na:
http://en.wikipedia.org/wiki/Ubuntu_One
- [18] Spletna stran storitve Ubuntu One. Dostopno 6.5.2013 na:
<https://one.ubuntu.com/>
- [19] Spletna enciklopedija Wikipedia – iskalni niz: »icloud«. Dostopno 7.5.2013 na:
<http://en.wikipedia.org/wiki/ICloud>
- [20] Spletna enciklopedija Wikipedia – iskalni niz: »amazon cloud drive«. Dostopno 11.5.2013 na:
http://en.wikipedia.org/wiki/Amazon_Cloud_Drive
- [21] Spletna stran za Google razvijalce. Dostopno 4.6.2013 na:
<https://developers.google.com/appengine/docs/whatisgoogleappengine>
- [22] Spletna stran ponudnika platforme Apprenda. Dostopno 8.6.2013 na:
<http://apprenda.com/platform/licensing/>