

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jakob Predin

**Oblačni informacijski sistem za podporo
delovanja zagonskih podjetij**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2018

© 2018, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

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

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko



Tematika naloge:

Kandidat naj v svojem delu predstavi problematiko informatizacije zagonskih podjetij. V nadaljevanju naj določi ključne funkcionalnosti tovrstnih sistemov ter jih implementira v oblačni obliki v okviru aktualnih razpoložljivih tehnologij.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Jakob Predin, Ljubljana, junij 2018.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jakob Predin

Oblačni informacijski sistem za podporo delovanja zagonskih podjetij

POVZETEK

Zagonsko podjetje je podjetje, ki na trg vstopa z novo inovativno rešitvijo in ustreznim poslovnim načrtom. Pri tem je ključno, da je zagonsko podjetje že od svojega začetka pri svojem poslovanju učinkovito in agilno, saj ima omejene vire financiranja. Ko takšno podjetje začne s poslovanjem, je v ospredju predvsem upravljanje odnosov s strankami podjetja ter pridobivanje investicij. Za vodenje teh je na voljo kar nekaj za to prilagojenih informacijskih sistemov. Namen pričujočega diplomskega dela je preučiti specifične delovanja zagonskih podjetij, ključnih funkcionalnosti različnih informacijskih sistemov, ki jih le-ta uporabljajo in olajšati njihovo poslovanje z uporabo namensko razvitega informacijskega sistema.

Ključne besede: informatizacija zagonskih podjetij, funkcionalnosti informacijskih rešitev, oblačna rešitev informacijskega sistema

University of Ljubljana
Faculty of Computer and Information Science

Jakob Predin

Cloud information system for startup operations

ABSTRACT

A startup is a company entering the market with a new and inovative solution with the appropriate business model. Due to limited funding it is essential that a startup is as effective and lean as possible in day to day operations from the beginning. When a startup is starting to operate the key focus is managing the relations with the client base, as well as getting funded. For managing these startups can use various customised information systems. The intent of the following paper is to study the specifics of startup operations, the key functionalities of information systems they use and easing their operations through the use of a purposely developed information system.

Key words: informatization of startups, features of information solutions, information system as a cloud solution

ZAHVALA

Zahvaljujem se svoji družini za njeno podporo in omogočanje študija, vsem, ki so me ob njem podpirali, doc. dr. Tomažu Hovelji za svetovanje ter svojemu mentorju prof. dr. Mihi Mrazu za vodenje, svetovanje in mentoriranje pri pisanju diplomske naloge.

— Jakob Predin, Ljubljana, junij 2018.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Motivacija	2
1.2 Struktura diplomskega dela	2
2 Opis stanja uporabe informacijskih sistemov v zagonskih podjetjih	5
2.1 Splošno o zagonskih podjetjih	5
2.1.1 Definicija zagonskega podjetja	6
2.1.2 Ključne aktivnosti zagonskih podjetij	7
2.1.3 Primeri zagonskih podjetij z inovativnimi rešitvami	8
2.1.4 Primeri zagonskih podjetij z inovativnim poslovnim modelom	8
2.2 Razlogi in viri financiranja zagonskih podjetij	8
2.2.1 Razlogi za pridobivanje investicij pri zagonskih podjetjih	8
2.2.2 Vrste investicij v zagonska podjetja	9
2.3 Pregled informacijskih sistemov za podporo poslovanja zagonskih podjetij	10
2.3.1 Pregled vrst informacijskih sistemov	10
2.3.2 CRM sistemi	11
2.3.3 Primer izbire in implementacije CRM sistema	12
3 Zahteve prototipa oblačnega sistema za podporo delovanja zagonskih podjetij	15
3.1 Analiza obstoječih sistemov na tržišču	16

3.1.1	Ključne funkcionalnosti CRM sistemov	16
3.1.2	Podrobnejša opredelitev ključnih funkcionalnosti prototipa sistema	17
3.2	Intervjuji s fokusno skupino angelskih investitorjev	20
3.2.1	Definicije zagonskega podjetja in razlogi za angelsko investiranje	21
3.2.2	Zanimive lastnosti zagonskega podjetja za angelsko investicijo	21
3.2.3	Nadzorovanje zagonskega podjetja po investiciji	21
3.2.4	Uporaba informacijskih sistemov pri nadzorovanju poslovanja	21
3.2.5	Cevovodni pogled v CRM sistemu	22
3.2.6	Pogled na opravila v CRM sistemu	22
3.2.7	Pogled na stike in potencialne stranke podjetja v CRM sistemu in sinhronizacija z drugimi storitvami	22
3.2.8	Pogled s pregledom aktivnosti in dogodkov v CRM sistemu	22
3.2.9	Druge ključne ali zaželjene funkcionalnosti CRM sistema	22
3.3	Verifikacija pridobljenih informacij z anketo pri izbranih zagonskih podjetjih	23
3.4	Ključne funkcionalnosti prototipa	25
3.4.1	Cevovodna funkcionalnost	25
3.4.2	Funkcionalnost s pogledom na opravila	25
3.4.3	Upravljanje s pogledom na stike in potencialne stranke podjetja	28
3.4.4	Funkcionalnost pregleda aktivnosti in dogodkov	28
3.4.5	Funkcionalnost vpogleda statistik uporabe sistema	28
4	Izdelava prototipa	31
4.1	Arhitektura informacijskega sistema	31
4.2	Uporabljene tehnologije	32
4.2.1	MongoDB	33
4.2.2	Express.js	33
4.2.3	React.js	34
4.2.4	Node.js	34
4.3	Predstavitev struktur oblačnega sistema	34
4.3.1	Predstavitev spletne aplikacije	35
4.3.2	Predstavitev strežniške aplikacije	37
4.3.3	Predstavitev implementacije podatkovne baze	40
4.4	Delovanje prototipa	41

4.4.1	Delovanje cevovodnega pogleda	42
4.4.2	Delovanje opravil	42
4.4.3	Delovanje upravljanja s stiki in potencialnimi strankami	43
4.4.4	Delovanje pogleda aktivnosti in dogodkov z opravi	43
4.4.5	Delovanje vpogleda statistik	45
4.5	Vzpostavitev oblačnega sistema	45
4.5.1	Vzpostavitev spletne in strežniške aplikacije	46
4.5.2	Vzpostavitev podatkovne baze	46
4.5.3	Zahteve oblačnega sistema	47
5	Sklepne ugotovitve in zaključek	49
A	Priloga	55
A.1	Končen izgled funkcionalnosti	55

1 Uvod

Zagonsko podjetje je termin za inovativno podjetje staro do 3 leta z manj kot 80 zaposlenimi in manj kot 20.000.000\$ letnih prihodkov [1]. Ta se v Sloveniji in po svetu poskušajo uveljavljati v praktično vsaki gospodarski panogi in stremijo h konstantnemu iskanju izboljšav in optimizacij obstoječih produktov in storitev na tržišču [2]. Zaradi dviga brezposelnosti, ki jo je povzročila finančna kriza iz leta 2008, je postala ustanovitev zagonskih podjetij zelo popularna [3]. Pri tem mora zaradi omejenih začetnih virov finančnih sredstev ter omejenih možnosti kreditiranja večina zagonskih podjetij iskati druge vire financiranja [4]. Proces iskanja teh virov je eden izmed ključnih poslovnih procesov zagonskega podjetja [2].

Ko zagonsko podjetje začne s poslovanjem, je poleg omenjenega procesa pridobivanja virov financiranja v ospredju predvsem upravljanje odnosov s strankami podjetja [2]. Za upravljanje teh lahko zagonska podjetja uporabljajo različne informacijske sisteme. Z njimi imajo možnost izvajanja poslovnih nalog, kot so prodaja, dodeljevanje opravil med zaposlenimi in iskanje virov financiranja. Pri tem nekatera zagonska podjetja naletijo na

določene probleme. Eden izmed poglavitnih je, da je zelo malo informacijskih sistemov prilagojenih za poslovanje zagonskih podjetij [5]. Nadalje strošek nakupa teh sistemov zagonskim podjetjem pogosto preprečuje njihovo uporabo pri poslovanju [5].

1.1 Motivacija

Skozi večletno sodelovanje s **Klubom Poslovnih angelov Slovenije** ter delom v slovenskem podpornem okolju za zagonska podjetja je avtor pričujoče naloge iz prve roke spoznal težave omenjene problematike pri izbiri informacijskih sistemov. Številna slovenska zagonska podjetja si pri svojem poslovanju želijo uporabljati nek informacijski sistem, vendar so obstoječe rešitve zaradi previsokih stroškov zanje velikokrat nedostopne. Za cilje diplomskega dela smo si zato zadali sledeče naloge:

- raziskati potrebe zagonskih podjetij;
- na podlagi ugotovitev definirati in overiti ključne funkcionalnosti sistema za upravljanje odnosov s strankami in pridobivanjem alternativnih virov financiranja zagonskih podjetij;
- na podlagi ključnih funkcionalnosti razviti prototip odprtokodnega sistema za podporo delovanja zagonskih podjetij;

1.2 Struktura diplomskega dela

Za doseganje zastavljenih ciljev je diplomsko delo strukturirano na naslednji način. V drugem poglavju je predstavljen pregled literature o zagonskih podjetjih, o virih financiranja za zagonska podjetja in literature o uporabljenih sistemih za vodenje poslovanja zagonskih podjetij. Namen pregleda je raziskati preučevano področje z namenom iskanja ključnih funkcionalnosti prototipa. V tretjem poglavju smo na podlagi pregleda literature iz drugega poglavja analizirali obstoječe informacijske sisteme na tržišču, ki jih uporabljajo zagonska podjetja. Na podlagi ugotovitev smo s tem definirali njihove ključne funkcionalnosti. Nato smo te funkcionalnosti overili najprej pri izbrani skupini angelskih investitorjev v zagonska podjetja, potem pa pri izbrani skupini zagonskih podjetij. Na podlagi pridobljenih informacij smo nato definirali ključne funkcionalnosti, ki bodo implementirane v prototipu ter način njihovega delovanja. V četrtem poglavju smo

na podlagi ključnih funkcionalnosti definiranih v tretjem poglavju opisali razvoj prototipnega oblachnega sistema. Pri tem smo se osredotočili na implementacijo ključnih funkcionalnosti, izbiro tehnologij za razvoj in ključne izzive pri razvoju. V petem poglavju smo povzeli celotno diplomsko delo in definirali predloge za izboljšave in nadgradnje prototipa.

2 Opis stanja uporabe informacijskih sistemov v zagonskih podjetjih

Literatura, pregledana v okviru diplomskega dela, je razdeljena na tri dele. Najprej je predstavljen pregled literature, ki definira zagonska podjetja in njihove ključne aktivnosti. Pregled je pomemben za seznanitev z naravo uporabnikov ciljnega prototipa in za iskanje že obstoječih informacijskih sistemov, ki podpirajo vodenje ključnih aktivnosti. Sledi pregled literature o virih financiranja zagonskih podjetij. Pregled je ključen za iskanje že obstoječih informacijskih sistemov, ki podpirajo enega izmed glavnih poslovnih procesov zagonskih podjetij [2]. Nato sledi še pregled literature o uporabljenih sistemih za vodenje poslovanja v zagonskih podjetjih. Namen pregleda je poiskati informacijske sisteme na podlagi ugotovitev iz prejšnjih dveh podpoglavij.

2.1 Splošno o zagonskih podjetjih

V sledečih razdelkih sledi pregled definicij in ključnih aktivnosti zagonskega podjetja.

2.1.1 Definicija zagonskega podjetja

Osnovna definicija zagonskega podjetja je zelo široka in ga definira kot vsako novo ustanovljeno podjetje. V praksi se ta definicija ne nanaša na podjetja, ki jih definiramo kot zagonska podjetja, zaradi česar se definicije zagonskega podjetja močno razlikujejo. Primer definicije iz začetka uvoda je samo ena izmed pogosteje uporabljenih, ki zagonsko podjetje definira na podlagi določenih lastnosti. Kljub razlikovanju definicij, v literaturi lahko med njimi najdemo nekatere skupne značilnosti [1]. Ker so potencialni uporabniki prototipa informacijskega sistema izključno zagonska podjetja, je potrebno ugotoviti te značilnosti ter definirati mejo, kdaj zagonsko podjetje ni več zagonsko podjetje. Sledi nekaj pogosto uporabljenih definicij zagonskega podjetja. Le te so sledeče:

- Steve Blank, profesor podjetništva na Univerzi Stanford, Berkeley, Columbia in NYU: *Zagonsko podjetje je organizacija ustvarjena za iskanje ponovljivega in skalabilnega poslovnega modela* [6].
- Eric Ries, ameriški podjetnik in avtor knjige "The Lean Startup": *Zagonsko podjetje je institucija ustvarjena pod pogoji ekstremne negotovosti za stvaritev novega produkta ali storitve* [7].
- Investopedia.com: *Zagonsko podjetje je podjetje, ki je v prvi fazi svojega poslovanja. Takšna podjetja so pogosto praviloma financirana s strani podjetnikov, ki so ustanovitelji podjetja, pri čemer poskušajo zaslužiti z razvojem produkta ali storitve za katero verjamejo, da zanjo na trgu obstaja določeno povpraševanje. Zaradi omejenih prihodkov ali visokih stroškov obratovanja večina izmed teh ni dolgoročno solventnih brez dodatnega financiranja s strani tveganega kapitala* [8].

Po pregledu definicij zagonskega podjetja smo pregledali tudi popularne definicije meje, kdaj zagonsko podjetje ni več zagonsko podjetje. Le te so sledeče:

- Alex Wilhelm, glavni urednik portala CrunchBase.com: *Če ima vaše podjetje pričakovanih 50 milijonov dolarjev prihodkov v naslednjih 12 mesecih, 100 ali več zaposlenih ali je vredno več kot 500 milijonov dolarjev, knjigovodsko ali drugače, morate obesiti vašo startup uniformo in doumeti, da ste samo eno izmed visokotehnoloških podjetij, ki sledi ali se aktivno izogiba prvi javni ponudbi delnic* [9].
- Paul Graham, ustanovitelj ameriškega pospeševalnika Y Combinator: *Podjetje je zagonsko podjetje dokler doživlja hitro rast* [10].

- Jay Habegger, izvršni direktor podjetja ownerIQ: *Zagonsko podjetje preneha biti zagonsko podjetje takrat, ko preneha inovirati* [11].

Za namene diplomskega dela bomo za definicijo zagonskega podjetja uporabili definicijo Erica Riesa, za opis meje, kdaj zagonsko podjetje ni več zagonsko podjetje, pa opis Alexa Wilhelma.

2.1.2 Ključne aktivnosti zagonskih podjetij

Zagonska podjetja se od velikih podjetij najbolj ločijo po tem, da na začetku delovanja še ne poznajo svojih ključnih kupcev [2]. Dokler zagonska podjetja zanimivosti svojih rešitev ne preverijo na trgu lahko samo predvidevajo, kdo so njihovi ključni kupci [2]. Zaradi tega in omejene količine finančnih sredstev je pomembno, da zagonsko podjetje pri poslovanju maksimizira učinkovitost. Za doseganje tega se večina zagonskih podjetij poslužuje t.i. "vitke metodologije" [12],[13]. Koncept vitkosti narekuje, da mora zagonsko podjetje od ustanovitve skrbno izbirati načine porabe svojih finančnih sredstev in pri tem biti čim bolj učinkovito [12]. To vključuje cenovno ugoden razvoj prvotne rešitve, iskanje cenovno najbolj ugodnih produktov in storitev potrebnih za poslovanje, učinkovito iskanje prodajnih kanalov in drugo [12]. Pomembno je tudi zavedanje, kdaj je potrebno spremeniti samo rešitev oz. produkt zagonskega podjetja (jo pivotirati), ter kdaj je ob neuspešnosti podjetja s poslovanjem potrebno končati [12]. Pristopi vitke metodologije vključujejo sledeča segmenta [13]:

- **Dobro poznavanje strank in trga:** Potrebno je prepoznati problem na tržišču, potencialne stranke za njegove rešitve, konkurenco ter konkurenčno prednost rešitve zagonskega podjetja.
- **Postopna adaptacija rešitve na podlagi odzivov trga:** Rešitev je potrebno spreminjati glede na odzive trga. Zagonsko podjetje to iteracijo ponavlja, dokler ne najde svojega mesta na trgu. Iz te točke se rešitev načeloma le še nadgrajuje, fokus pa postane iskanje novih strank in razvoj novih rešitev.

Zagonsko podjetje torej išče *inovativno rešitev* ter način njene realizacije, ki bi svoje mesto z rešitvijo nekega problema našla na tržišču. Ta pristop zahteva dobro poznavanje potencialnih strank, kar je koristno tako pri procesu prodaje, kot tudi pri adaptaciji rešitve. Pri tem je lahko rešitev zagonskega podjetja inovativna zaradi same rešitve ali poslovnega modela, s katerim jo ponujajo [12].

2.1.3 Primeri zagonskih podjetij z inovativnimi rešitvami

Ta zagonska podjetja na trgu izstopajo predvsem zaradi inovativnosti produkta oz. inovativnosti storitve, ki jo ponujajo [12]. Primeri teh so sledeči:

- Twitter: socialno omrežje, na katerem lahko uporabniki objavijo vsebino z omejenim številom znakov;
- Github: skupna kolaboracija in varnostno kopiranje pri razvoju programske kode;
- Snapchat: socialno omrežje, kjer uporabniki med sabo delijo medijske vsebine, ki se po določenem času izbrišejo uporabnikove iz naprave;

2.1.4 Primeri zagonskih podjetij z inovativnim poslovnim modelom

Ta zagonska podjetja na trgu izstopajo predvsem zaradi njihovega inovativnega poslovnega modela [12]. Primeri teh so sledeči:

- DollarShaveClub: strankam enkrat na mesec na dom pošljejo britvice (na podlagi mesečnih naročnin);
- Netflix: neomejeno brskanje na portalu ter dostop do video vsebin (na podlagi mesečnih naročnin);
- Uber: z deljenimi vožnjami lahko uporabniki ceneje kot s taksiji potujejo na kratke razdalje;

2.2 Razlogi in viri financiranja zagonskih podjetij

V pregledu literature o virih financiranja za zagonska podjetja izvedemo pregled razlogov, zakaj zagonsko podjetje potrebuje investicije in pregled vrst investicij v zagonska podjetja.

2.2.1 Razlogi za pridobivanje investicij pri zagonskih podjetjih

Investicije v zagonska podjetja so nujne za njihovo preživetje, saj na začetku poslovanja nimajo zadostne količine finančnih sredstev [2],[4]. Zaradi tega ter omejenih možnosti kreditiranja mora večina izmed teh iskati druge vire financiranja [4]. Te vključujejo lastniško financiranje in investicije z uporabo množičnega financiranja [14]. V zameno za prejem dodatnih sredstev se ustanovitelji odrečejo deležu v podjetju oz. se zavežejo

izpolniti določene pogoje [4]. Višina investicije je pri tem praviloma odvisna od vrste investicije [4]. Razlogi, zakaj zagonska podjetja potrebujejo investicije, so lahko sledeči [4],[15],[16],[17]:

- za dokončanje razvoja produkta,
- za razvoj novega produkta,
- zaradi poslovanja z izgubo,
- zaradi želje po širitvi na tuje trge,
- zaradi želje po masovni proizvodnji produkta,
- zaradi pospeševanja procesa prodaje,
- za izvedbo marketinških kampanj,
- za pridobitev strateškega partnerja zagonskega podjetja skozi investicijo,
- zaradi možnosti aktivnega testiranja ustreznosti rešitve na tržišču.

2.2.2 Vrste investicij v zagonska podjetja

Za iskanje ključnih funkcionalnosti sistema smo raziskali najpogostejše vrste investicij v zagonska podjetja. Te so sledeče [4]:

- **FFF investicije:** Zagonska podjetja prejmejo FFF investicije v začetnih fazah delovanja [17],[18]. Takrat imajo ponavadi razvito samo idejo o rešitvi nekega problema [17]. Kratica (FFF) v angleščini pomeni družino, prijatelje in bedake. To nakazuje skupino investorjev, ki podjetniško ekipo poznajo in v njihovo idejo verjamejo zaradi podjetnikov samih in ne zaradi odziva iz tržišča [17],[18].
- **Angelske investicije:** Zagonska podjetja prejmejo angelske investicije v fazi med FFF investicijami in VC (angl. *venture capital*) investicijami [16]. Čeprav ima v tej fazi zagonsko podjetje tipično že razvit najmanjši sprejemljivi produkt, je verjetnost neuspeha še vedno visoka [12],[16]. Angelski investitorji so pri tem individualni privatni investitorji, ki investirajo neposredno v zasebna podjetja, pri katerih nimajo družinskih ali prijateljskih povezav [16].

- **VC investicije:** Zagonska podjetja prejmejo VC investicije v fazi, ko je pričakovana dolgoročna rast zagonskega podjetja [19]. Za možnost prejema te investicije mora imeti zagonsko podjetje visok potencial rasti, veliko potencialno tržišče, visoke potencialne prihodke in zmožnost kasnejše prodaje podjetja [20]. VC investitorji so pri tem privatni investitorji, investicijske banke in druge finančne institucije [19].
- **Investicije preko množičnega financiranja:** Množično financiranje je spletno distribuiran model investiranja, kjer množica ljudi z relativno majhno denarno investicijo pomaga zagonskemu podjetju pridobiti nujno potrebna finančna sredstva [4]. Pri tem se višina nujno potrebnih finančnih sredstev od podjetja do podjetja razlikuje [4]. V zameno za denarno investicijo lahko posameznik prejme končni produkt, obresti od investicije, sredstva v naravi, delež v podjetju itd [4].

2.3 Pregled informacijskih sistemov za podporo poslovanja zagonskih podjetij

Sledi pregled literature o informacijskih sistemih, ki jih zagonska podjetja uporabljajo pri svojem poslovanju ter primer izbire in implementacije enega izmed omenjenih sistemov.

2.3.1 Pregled vrst informacijskih sistemov

Visokotehnološka zagonska podjetja lahko za vodenje svojega poslovanja uporabljajo večje število različnih informacijskih sistemov. Ti lahko vključujejo sisteme za upravljanje tekstovnih dokumentov, sisteme za vodenje projektov, sisteme za vodenje odnosov s strankami itd. Zaradi stroškovnih omejitev je pomembno, da zagonsko podjetje uporablja samo informacijske sisteme, ki časovno ali stroškovno pozitivno vplivajo na njihovo delovanje [12],[21]. Potrebni informacijski sistemi se razlikujejo glede na panogo, v kateri deluje zagonsko podjetje [21].

Informacijski sistemi, ki lahko izboljšajo poslovanje zagonskih podjetij, vključujejo sisteme za izvajanje sledečih funkcij [22]:

- vodenje računovodstva,
- objavljanje vsebin,
- vodenje odnosov s strankami,

- upravljanje podatkov v podatkovnih bazah,
- vodenje spletne prodaje,
- vodenje delovnega časa.

2.3.2 CRM sistemi

Za upravljanje odnosov s strankami ali pridobivanje investicij zagonska podjetja uporabljajo CRM sisteme (angl. *Customer Relationship Management*) [5]. Ker je definiranje njihovih ključnih funkcionalnosti eden izmed ciljev diplomskega dela, se nadaljnja obravnava osredotoča nanje.

Upravljanje odnosov s strankami izhaja iz konceptov marketinga odnosov [5]. CRM sistemi te koncepte dodelajo in definirajo pristope sodelovanja s strankami v procesih prodaje, marketinga in podpore [5]. Pri tem CRM sisteme delimo na sledeče vrste [5]:

- tradicionalne CRM sisteme,
- elektronske CRM sisteme (eCRM),
- mobilne CRM sisteme (mCRM),
- socialne CRM sisteme (sCRM).

Največji ponudniki CRM sistemov so pri tem sledeči [23]:

- Salesforce,
- SAP,
- Oracle,
- Microsoft.

Pogosto se uporabljajo tudi odprtokodne različice sistemov CRM različnih ponudnikov. Nekatere izmed teh vključujejo:

- Suite CRM,
- Civi CRM,

- Vtiger.

Funkcionalnosti in primerjave sistemov CRM so podrobno opisane v tretjem poglavju skozi analizo obstoječih sistemov na tržišču.

2.3.3 Primer izbire in implementacije CRM sistema

Zagonska podjetja lahko informacijske sisteme izbirajo po številnih metodologijah. Ena izmed bolj uveljavljenih je metodologija avtorjev Anil S. Jadhava in Rajendra M. Sonar [24]. Njuna metodologija definira postopek izbire informacijskega sistema na osnovi primerjave številnih drugih metodologij izbire informacijskih sistemov. Metodologija vključuje naslednje korake:

1. *Ugotovi, zakaj obstaja potreba za nakup nekega informacijskega sistema in kdo vse ponuja takšne rešitve.* Da lahko zagonsko podjetje izbere CRM sistem mora sprva interno definirati, katere so za njih ključne funkcionalnosti. Zagonsko podjetje nato začne z iskanjem ponudnikov CRM sistemov.
2. *Iz seznama ponudnikov izberi tiste, katerih informacijski sistem se ti zdi zadovoljiv.* Po iskanju zagonsko podjetje iz seznama ponudnikov sistemov izbere tiste, pri katerih se jim zdi njihov CRM sistem zadovoljiv.
3. *Iz dobljenega seznama ponudnikov rešitev odstrani tiste, ki ne podpirajo zahtevanih funkcionalnosti.*
4. *S poljubno metodologijo empirično med sabo ovrednoti ostale rešitve.* Končen seznam ponudnikov CRM sistemov mora zagonsko podjetje empirično ovrednotiti med seboj. Te lahko primerjajo s številnimi metodologijami, ki se osredotočajo na aspekte kot so strošek nakupa sistema, dodatne funkcionalnosti, prijaznost uporabniškega vmesnika itd.
5. *Pred nakupom stestiraj najboljše ovrednoteno rešitev v primernem okolju.*
6. *S ponudnikom definiraj pogodbo, način in višino plačila, število licenc in način podpore po nakupu.* V kolikor CRM sistem tudi po testiranju ustreza zahtevam potem zagonsko podjetje s ponudnikom sklene pogodbo in izvrši plačilo.

7. *Po nakupu izbrani sistem integriraj v svoje poslovanje.* Po nakupu zagonsko podjetje CRM sistem implementira v svoj prodajni proces in začne z uporabo.

3 Zahteve prototipa oblačnega sistema za podporo delovanja zagonskih podjetij

Za ustrezen zajem zahtev je potrebna analiza obstoječih informacijskih sistemov na tržišču, s katerimi zagonska podjetja vodijo odnose s strankami in pridobivajo investicije. Tipično se za to uporabljajo CRM sistemi [5]. Zaradi tega je smiselno seznam ključnih funkcionalnosti opredeliti s pomočjo analize funkcionalnosti obstoječih CRM sistemov na tržišču. Nadalje je potrebno ugotoviti, katere ključne funkcionalnosti zagonska podjetja zares nujno potrebujejo.

Po analizi obstoječih CRM sistemov je za dodatno potrditev izbranih ključnih funkcionalnosti smiselno opraviti intervjuje s fokusno skupino angelskih investorjev. Namen tega je ugotoviti, katere funkcionalnosti investorji v zagonska podjetja smatrajo za ključne, saj so njihove investicije pogosto vezane na uporabo nekega CRM sistema [5]. Tako pridobljene informacije je smiselno preveriti še pri zagonskih podjetjih, kjer podjetniki sami s svojimi izkušnjami povedo, katere so za njih ključne funkcionalnosti.

3.1 Analiza obstoječih sistemov na tržišču

Kot omenjeno se bo začetna obravnava fokusirala na CRM sisteme. Sledi podroben pregled funkcionalnosti in razlikovanj omenjenih sistemov.

3.1.1 Ključne funkcionalnosti CRM sistemov

Na tržišču je na voljo precej različnih CRM sistemov, ki implementirajo različne funkcionalnosti. Ker vsak ponudnik CRM sistemov izpostavlja različen nabor ključnih funkcionalnosti, je smiselno primerjati, katere so implementirane v večjih CRM sistemih. Za primerjavo ključnih funkcionalnosti smo izbrali CRM sisteme treh največjih ponudnikov na globalnem trgu [23]. Ti vključujejo sisteme "Salesforce Sales Cloud", "SAP CRM for Sales" ter "Oracle CRM". Pri vsakem smo zabeležili ključne funkcionalnosti, kot jih je definiral ponudnik sam. Nato smo preverili, katere izmed teh funkcionalnosti so implementirane v posameznih testiranih CRM sistemih. V naš seznam ključnih funkcionalnosti smo vključili samo tiste, ki so podprte v vseh treh testiranih CRM sistemih. Te so podrobno opisane v naslednjem razdelku. Rezultati primerjave so predstavljeni v tabeli 3.1. Ker se poimenovanja posameznih funkcionalnosti lahko razlikujejo med sistemi, sledijo kratki opisi vseh funkcionalnosti. Omenjene funkcionalnosti so sledeče [25],[26],[27]:

- **Upravljanje s stiki in potencialnimi strankami:** funkcionalnost omogoča uporabniku vpogled v podatke posameznega stika in potencialne stranke, možnost njihovega urejanja, izvajanje akcij nad njimi itd;
- **Upravljanje s priložnostmi:** funkcionalnost omogoča uporabniku vpogled v vse in posamezne priložnosti oz. potencialne posle podjetja, možnost njihovega urejanja, izvajanje akcij nad njimi itd;
- **Upravljanje z opravili:** funkcionalnost omogoča uporabniku prikaz seznama njegovih opravil, možnost njihovega posameznega ogleda, izvajanja akcij nad njimi itd;
- **Pregled aktivnosti in dogodkov:** funkcionalnost omogoča uporabniku vpogled v vse njegove aktivnosti, ki lahko vključujejo dogodke ali naloge;
- **Analitika in poročanje:** funkcionalnost omogoča uporabniku vpogled v statistične podatke, ki se generirajo skozi uporabo CRM sistema in pridobivanje obvestil v odvisnosti od njihovih vrednosti;

- **Uporaba preko mobilne aplikacije:** funkcionalnost omogoča uporabniku dostop do CRM sistema preko mobilnih aplikacij na sistemih kot so iOS ali Android;
- **Avtomatizacija prodaje:** funkcionalnost omogoča vršenje vnaprej določenih akcij nad stiki, potencialnimi strankami, priložnostmi ali nalogami, v kolikor so ureničeni za to določeni pogoji;
- **Integracije z drugimi storitvami:** funkcionalnost omogoča integracije z drugimi storitvami kot so ponudniki e-pošte, ponudniki množičnega pošiljanja e-poštnih sporočil, socialnimi omrežji itd;
- **Upravljanje odnosov s partnerji podjetja:** funkcionalnost je implementirana kot poseben del CRM sistema, kjer se odnosi s partnerji podjetja vodijo podobno kot odnosi s potencialnimi strankami;
- **Učenje uporabe sistema z ročno pripravljenimi primeri:** funkcionalnost omogoča uporabniku učenje uporabe sistema z ročno pripravljenimi primeri posameznega podjetja;

3.1.2 Podrobnejša opredelitev ključnih funkcionalnosti prototipa sistema

Kot omenjeno bomo za ključne funkcionalnosti izbrali tiste, ki so popolnoma podprte v vseh testiranih CRM sistemih predstavljenih v tabeli 3.1. Da so te funkcionalnosti ustrezne za zagonska podjetja, smo v nadaljevanju potrdili z intervjuji ter anketo. V nadaljevanju sledijo podrobni opisi in način delovanja ključnih funkcionalnosti, ki bodo implementirane v prototipu sistema za podporo delovanja zagonskih podjetij. Ti opisi so sledeči:

- **Upravljanje s stiki in s potencialnimi strankami:** funkcionalnost je osnova vsakega CRM sistema, ki se skupaj z funkcionalnostjo upravljanja z opravili najpogosteje uporablja [28]; njeno bistvo je, da omogoča uporabniku pridobivanje in upravljanje z informacijami o posameznem stiku ali potencialni stranki; v danem trenutku so uporabniku o stiku ali potencialni stranki lahko na voljo sledeči podatki:
 - kontaktni podatki in splošne informacije,
 - kje v prodajnem procesu se nahaja,

Funkcionalnosti	Salesforce Sales Cloud	SAP CRM for Sales	Oracle CRM
Upravljanje s stiki in potencialnimi strankami	Da	Da	Da
Upravljanje s priložnostmi	Da	Da	Da
Upravljanje z opravili	Da	Da	Da
Pregled aktivnosti in dogodkov	Da	Da	Da
Analitika in poročanje	Da	Da	Da
Uporaba preko mobilne aplikacije	Da	Ne	Ne
Avtomatizacija prodaje	Da	Ne	Da
Upravljanje z naročili	Da	Ne	Da
Integracije z drugimi storitvami	Veliko podprtih, vendar so nekatere plačljive	Samo e-pošta	Samo e-pošta
Upravljanje odnosov s partnerji podjetja	Ne	Ne	Da
Učenje uporabe sistema z ročno pripravljenimi primeri	Ne	Ne	Da

Tabela 3.1 Primerjava implementacij ključnih funkcionalnosti v CRM sistemih.

- katere so bile zadnje interakcije,
 - katera so bila zadnja opravila,
 - pomembne statistike in datoteke.
- **Upravljanje s priložnostmi:** funkcionalnost zajema pregled vseh priložnosti (v nadaljevanju poslov) in možnost vpogleda v vsako od njih; v implementaciji te funkcionalnosti se lahko različni CRM sistemi med seboj precej razlikujejo; nekateri se pri tem osredotočajo na posle izključno s stališča verjetnosti posameznega posla, nekateri iz pozicije posla v prodajnem procesu, nekateri pa na kombinacijo obeh; eden izmed načinov je predstavitev upravljanja potencialnih poslov s funkcionalnostjo cevovodnega pogleda; ta omogoča sledeče funkcionalnosti:
- prenašanje poslov iz ene faze v drugo;
 - določitev, ali je posel dobljen ali izgubljen;
 - poimenovanje posamezne faze prodajnega procesa;

drugi del funkcionalnosti je implementacija vpogleda v posamezen posel; tudi v tem primeru se načini implementacije močno razlikujejo med sistemi, vendar pa je pri vpogledu običajno prikazano sledeče:

- ime posla;
- vrednost posla;
- verjetnost posla;
- rok izvedbe posla;

ideja je, da se za posamezen potencialen posel skozi uporabo dodajajo podatki kot so kdaj se je posel premaknil iz ene v drugo fazo prodajnega procesa, sprememba verjetnosti ali vrednosti posla, dodajanje dodatnih informacij o poslu, pregled števila oseb, ki so bile zadolžene za posel itd;

- **Upravljanje z opravili:** pri tej funkcionalnosti CRM sistemov so uporabniku prikazana razvrščena opravila, ki jih je potrebno izvršiti pri posameznem poslu, potencialni stranki ali stiku; opravila običajno vsebujejo sledeče dejavnosti:
- klice;

- pošiljanje e-sporočil;
- sestanke;
- interna opravila;

opravila so lahko sortirana na sledeče načine:

- časovno sortiranje;
 - prioritarno sortiranje;
 - abecedno sortiranje;
- **Pregled aktivnosti in dogodkov:** funkcionalnost, ki je pogosto poimenovana "koledar", vključuje dnevni, tedenski in mesečni pogled vseh dogodkov in aktivnosti posameznega uporabnika; pri tem k aktivnostim štejemo opravila uporabnika v CRM sistemu, k dogodkom pa opravila nepovezana s CRM sistemom kot so npr. poslovna potovanja. CRM sistemi se pogosto ločijo po tem, ali v tem vpogledu vključujejo samo opravila ali dogodke in opravila; funkcionalnost se med drugim uporablja za preverjanje zasedenosti in planiranje prihodnjih opravil in dogodkov;
 - **Analitika in poročanje:** funkcionalnost omogoča vpogled v uspešnost prodaje podjetja; z njo se lahko preveri uspešnost prodajnikov glede na predhodne podatke in cilje iz poslovnega načrta; s tem se lahko ugotovi, v katerem procesu podjetje posluje najboljše in v katerem najslabše; elementi, ki so vključeni v to funkcionalnost, se lahko močno razlikujejo med sistemi; ti lahko vključujejo sledeče podatke:
 - število pridobljenih potencialnih poslov;
 - število nerealiziranih potencialnih poslov;
 - število opravljenih klicev, poslanih e-poštnih sporočil in opravljenih sestankov;
 - število na novo ustvarjenih potencialnih poslov;

3.2 Intervjuji s fokusno skupino angelskih investitorjev

Kot omenjeno so angelski investitorji pogosto razlog, da zagonsko podjetje po prejemu investicije začne uporabljati nek CRM sistem [5]. Posledično je smiselna seznanitev z njihovim pogledom na zagonska podjetja, investicije v njih in zanje ključne funkcionalnosti CRM sistemov. Intervjuji so bili izvedeni s tremi slovenskimi angelskimi investitorji,

ki aktivno delujejo v podpornem okolju za zagonska podjetja. Ti so bili izbrani zaradi njihovih izkušenj pri vodenju lastnih podjetij ter zagonskih podjetij po prejemu njihove investicije. Področja pogovora so bila izbrana za namene pridobivanja njihove definicije ter pogleda na zagonska podjetja in za potrditev ključnih funkcionalnosti pridobljenih skozi analizo CRM sistemov. V nadaljevanju sledi analiza njihovih odgovorov.

3.2.1 Definicije zagonskega podjetja in razlogi za angelsko investiranje

Angelski investitorji so zagonska podjetja definirali kot mlada podjetja z ekipo sposobnih ljudi, ki imajo rešitve ali ideje, ki izhajajo iz realnega sveta. Kot razloge, zakaj so postali angelski investitorji, so navedli radovednost, željo po kreaciji skozi finančno podporo zagonskim podjetjem ter oplemenitenju njihovega denarja in možnost učinkovitega spremljanja najnovejših trendov.

3.2.2 Zanimive lastnosti zagonskega podjetja za angelsko investicijo

Angelski investitorji so kot zanimive lastnosti izpostavili vrsto panoge zagonskega podjetja, socialni vidik njihove rešitve (če ima neko zagonsko podjetje potencial pozitivnih sprememb na celotno družbo) in ekonomske kazalnike (rast zagonskega podjetja, najdeno mesto na tržišču itd.). Pomembno je, da zagonska podjetja ponujajo nišno rešitev, ki je zaželeno, ima internacionalne aspiracije in je inovativna tudi takrat, ko je v skladu s pravom.

3.2.3 Nadzorovanje zagonskega podjetja po investiciji

Angelski investitorji so poudarili, da je potrebno jasno določiti poslovne cilje in imeti nadzor nad poslovnimi procesi. Nadalje je potrebno določiti (ali uporabiti) sistem za vpogled nad temi procesi.

3.2.4 Uporaba informacijskih sistemov pri nadzorovanju poslovanja

Angelski investitorji so izpostavili nujnost uporabe informacijskega sistema za nadzorovanje poslovanja. V idealnem primeru naj bi se za to uporabljal CRM sistem. V kolikor tega zagonsko podjetje ne more pridobiti, morajo uporabljati vsaj nek sistem za upravljanje z razpredelnicami kot sistem Microsoft Excel. To je ključno za jasen pregled nad številom strank, segmentacijo trgov in stanje prodajnega procesa.

3.2.5 Cevovodni pogled v CRM sistemu

Angelski investitorji trdijo, da je način segmentacije, kot je cevovodni pogled, nujen del CRM sistema. Vedno je potrebno biti seznanjen s potencialnimi kupci ter z njihovim mestom v prodajnem procesu. Nujna je tudi jasnost mesta pridobivanja ter izgube strank.

3.2.6 Pogled na opravila v CRM sistemu

Angelski investitorji so izpostavili, da je nek način za vodenje in izvajanje opravil ključen, da pa CRM sistemi za to niso nujno najboljše orodje. Pri zagonskem podjetju je pomembno, da za to uporabljajo CRM sistem, zato da lahko zaposleni izvajajo naloge in da imajo hkrati investitorji vpogled nad dogajanjem v zagonskem podjetju.

3.2.7 Pogled na stike in potencialne stranke podjetja v CRM sistemu in sinhronizacija z drugimi storitvami

Angelski investitorji so poudarili nujnost pogleda na stike in potencialne stranke. S tem se lahko lažje segmentira kupce, hitro opravlja klice in razpolaga s podatki o stiku. To postane še posebej pomembno, ko ima zagonsko podjetje veliko stikov.

3.2.8 Pogled s pregledom aktivnosti in dogodkov v CRM sistemu

Angelski investitorji trdijo, da je pogled na aktivnosti in dogodke nujen v CRM sistemu. Uporabniki morajo vedno imeti možnost videti svoj osebni koledar, svoje naloge, prihajajoče aktivnosti in podobne aktivnosti vezane na dejavnosti znotraj podjetja.

3.2.9 Druge ključne ali zaželjene funkcionalnosti CRM sistema

Angelski investitorji so za druge ključne funkcionalnosti navedli željo po integraciji sistema s storitvami kot so Google Apps in kreacijo mesta za shranjevanje podatkov in datotek vezane na določeno stranko. Navedene zaželjene funkcionalnosti sistema so bile sortiranje podatkov kot so številni statistični pogledi (število potencialnih strank v cevovodu, pričakovani prihodki v nekem časovnem obdobju itd.) in možnost zajema podatkov o nekem stiku iz čim večjega števila virov, kot so socialna omrežja.

Ključna funkcionalnost \ Potrebnost	Da	Ne
Cevovodni pogled	80%	20%
Pogled na opravila	90%	10%
Pogled na stike in potencialne stranke podjetja	100%	0%
Pogled s pregledom aktivnosti in dogodkov	80%	20%
Integracije z drugimi storitvami	80%	20%

Tabela 3.2 Rezultati ankete o ključnih funkcionalnostih v CRM sistemih.

3.3 Verifikacija pridobljenih informacij z anketo pri izbranih zagonskih podjetjih

Po zajemu zahtev skozi analizo CRM sistemov in intervjuji s skupino izbranih angelskih investitorjev so bile pridobljene informacije preverjene tudi pri zagonskih podjetjih z anketo. Izbranih je bilo 10 zagonskih podjetij, ki ponujajo računalniško omogočene rešitve in so po definiciji Alexa Wilhelma [9] še lahko smatrana kot zagonska podjetja. Tem so bila zastavljena vprašanja o ključnih in zaželenih funkcionalnostih v CRM sistemih. Pri tem so bila zastavljena vprašanja izbrana na podlagi ugotovitev analize CRM sistemov in intervjujev z angelskimi investitorji. Sledijo njihovi odgovori prikazani v tabelah 3.2, 3.3 in 3.4 z dodanimi komentarji.

Iz tabele 3.2 je razvidno, da večina vprašanih meni, da je cevovodni pogled potreben del sistema CRM. Ker je ta funkcionalnost potrebna pri uporabi drugih funkcionalnosti je implementirana tudi v končni prototip sistema. Večina vprašanih prav tako meni, da so pogled na opravila, pogled na stike in potencialne stranke podjetja in pogled s pregledom aktivnosti in dogodkov ključne funkcionalnosti CRM sistemov. Ker so našete funkcionalnosti razbrane kot ključne tudi v prejšnjih dveh razdelkih so implementirane v končnem prototipu sistema. Tudi pri funkcionalnosti integracij z drugimi storitvami večina vprašanih meni, da so te ključne pri sistemih CRM. Posledično je prototip sistema optimiziran za kasnejše možne integracije.

Rezultati ankete o drugih ključnih funkcionalnostih CRM sistemov so razvidni v tabeli 3.3. V prototip sistema je zaradi večkratne omembe dodan statistični modul, v osnovno

Druge ključne funkcionalnosti CRM sistema \ Potrebnost	Da	Ne
Opomniki	10%	90%
Statistični modul	10%	90%
Povezanost med funkcionalnostmi sistema	10%	90%
Časovnica posameznega projekta oz. posla	10%	90%
Kompleten pregled aktivnosti pri posamezni stranki	10%	90%

Tabela 3.3 Rezultati ankete o drugih ključnih funkcionalnostih CRM sistemov.

Zaželjene funkcionalnosti sistema CRM \ Potrebnost	Da	Ne
Sinhronizacija podatkov z drugimi storitvami (npr. Google Stiki)	40%	60%
Različne statistike (npr. vrednost vseh aktivnih potencialnih poslov v sistemu)	90%	10%
Uporaba e-poštnih storitev direktno iz CRM sistema	80%	20%
Integracija s sistemom Mailchimp	70%	30%
Možnost spremljanja informacij iz socialnih profilov (Facebook, Twitter itd.)	60%	40%
Možnost upravljanja z dokumenti direktno iz CRM sistema	50%	50%
Povezljivost z aplikacijami za klicne centre	10%	90%
Integracija s telefonom	10%	90%

Tabela 3.4 Rezultati ankete zaželenih funkcionalnosti CRM sistemov.

funkcionalnost pa je vključena tudi povezanost med različnimi funkcionalnostmi in kompleten pregled aktivnosti pri posamezni stranki. Opomniki so v prototipu tretirani kot opravila, časovnica pri posameznem projektu pa ostaja kot možna funkcionalnost za integracijo v naslednjih verzijah. Pri izvedbi ankete je potrebno poudariti, da omenjene druge ključne funkcionalnosti niso bile navedene na sami anketi in da je bila vsaka posamezna funkcionalnost navedena s strani le enega anketiranega zagonskega podjetja.

Rezultati ankete željenih funkcionalnosti CRM sistemov so prikazani v tabeli 3.4. Z izjemo statističnega modula, ki je že izbran kot funkcionalnost prototipa, so našete funkcionalnosti lahko kasnejše nadgradnje prototipa.

3.4 Ključne funkcionalnosti prototipa

Glede na analizo obstoječih sistemov, pogovora z angelskimi investitorji in odgovorov, pridobljenih neposredno s strani zagonskih podjetij, je moč definirati ključne funkcionalnosti prototipa. Te so sledeče:

- cevovodna funkcionalnost z upravljanjem poslov v prodajnem procesu,
- funkcionalnost z opravili, vezanimi na stike in potencialne stranke ter posle,
- funkcionalnost upravljanja s stiki in potencialnimi strankami,
- funkcionalnost pregleda aktivnosti in dogodkov (koledar),
- funkcionalnost ogleda izbranih statistik skozi uporabo sistema.

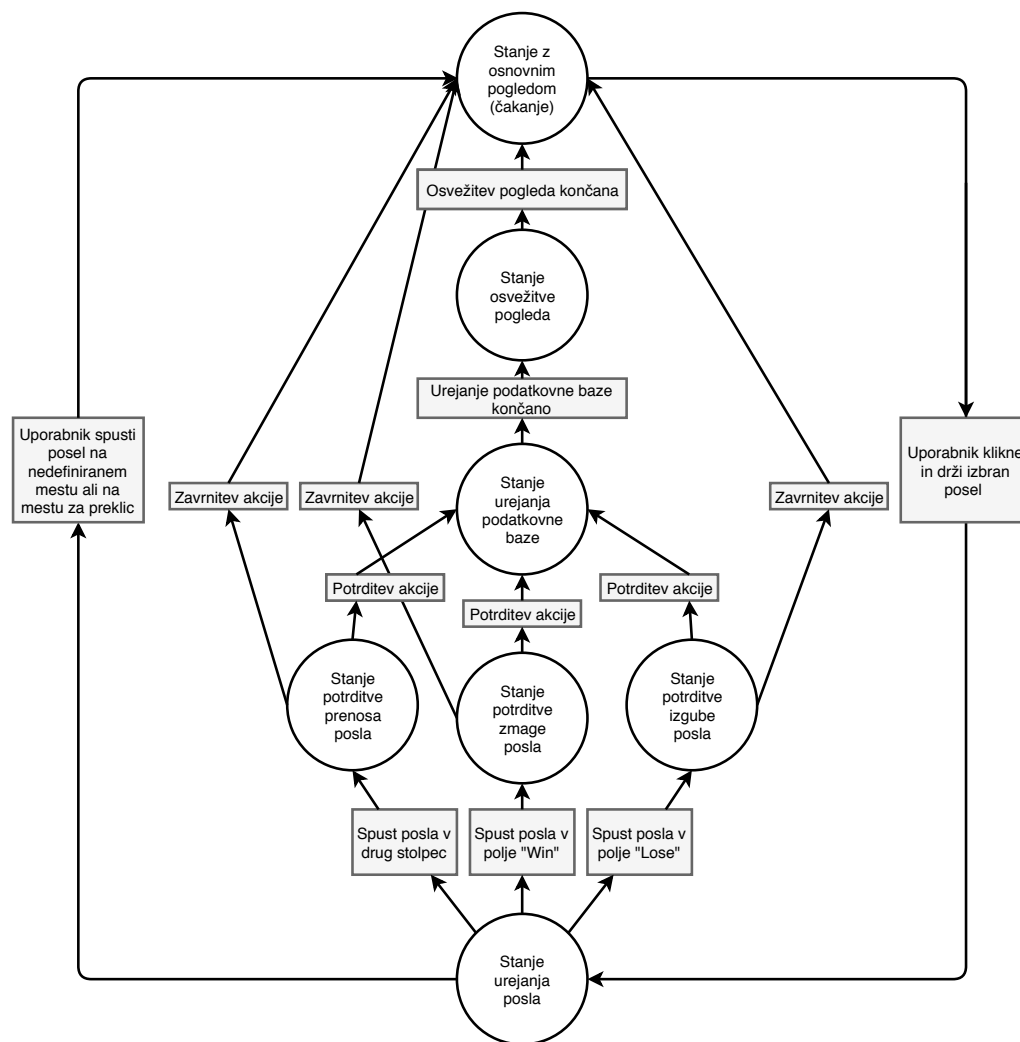
V nadaljevanju je po vzoru Petrijevih mrež [29] predstavljeno delovanje vseh ključnih funkcionalnosti, ki so implementirane v prototipu, pri čemer je končen izgled ključnih funkcionalnosti predstavljen v naslednjem poglavju.

3.4.1 Cevovodna funkcionalnost

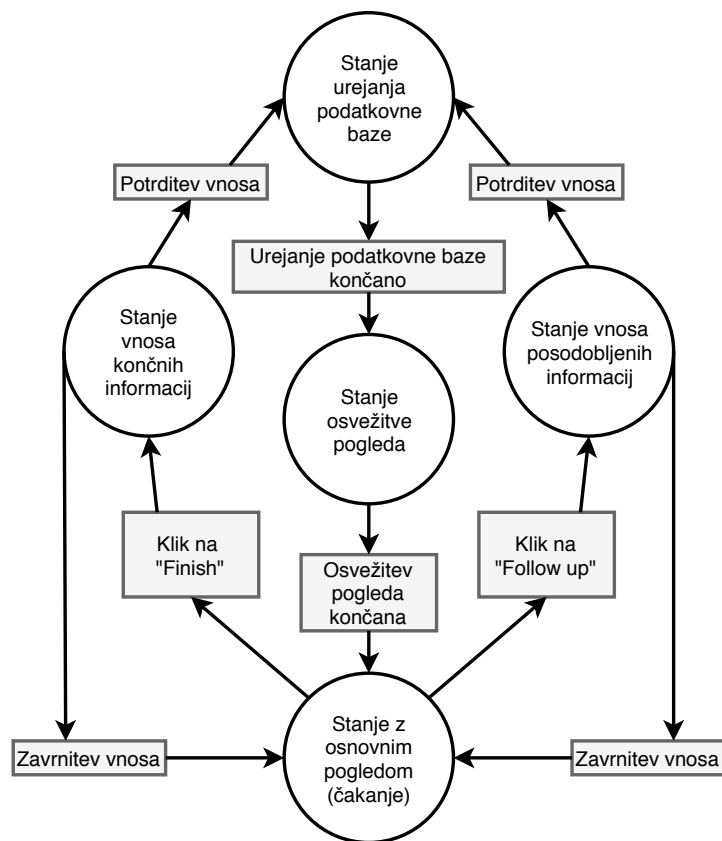
Funkcionalnost upravljanja s posli je zajeta v cevovodni funkcionalnosti poimenovani "Pipeline". V njej so v prototipu posli prikazani kot polja, ki so zajeti v posameznih fazah cevovodnega pogleda. Posli se bodo prikazali v delu cevovodnega pogleda, ki je slika prodajnega procesa, kamor pripadajo. Novi posli se dodajajo preko fiksno pozicioniranega gumba. Delovanje funkcionalnosti je ponazorjeno s Petrijevo mrežo na sliki 3.1.

3.4.2 Funkcionalnost s pogledom na opravila

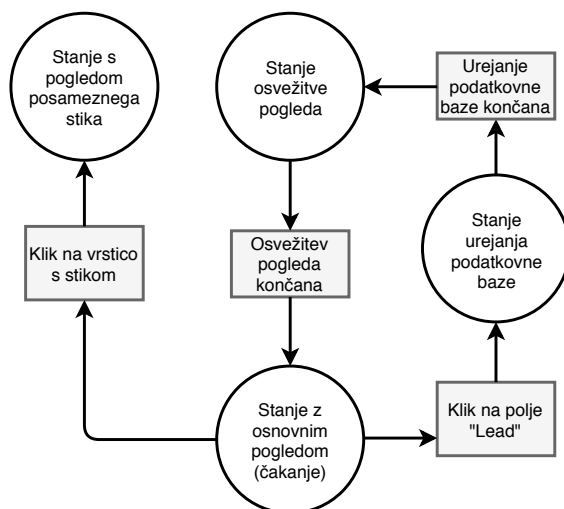
Funkcionalnost s pogledom na opravila vezana na stike in potencialne stranke ter posle je zajeta v funkcionalnosti poimenovani "Tasks". Opravila (v prototipu sistema prikazana kot vrstice tabele) so prikazana z večimi različnimi kriteriji. Vsebujejo ime opravila, na koga je opravilo vezano, ponazoritev posamezne naloge z ikono, rok za izvedbo opravila in pomembnost opravila. Dodana sta dva gumba. S klikom na prvega se posamezno opravilo lahko še nadaljuje in posodobi, če še ni končano oz. se morajo dodati novi podatki, s klikom na drugega pa se opravilo konča. Nova opravila se dodajajo preko fiksno pozicioniranega gumba. Delovanje funkcionalnosti je ponazorjeno s Petrijevo mrežo na sliki 3.2.



Slika 3.1 Prikaz delovanja cevovodne funkcionalnosti s Petrijevo mrežo.



Slika 3.2 Prikaz delovanja funkcionalnosti s pogledom na opravila s Petrijevo mrežo.



Slika 3.3 Prikaz delovanja funkcionalnosti s pogledom na stike in potencialne stranke podjetja s Petrijevo mrežo.

3.4.3 Upravljanje s pogledom na stike in potencialne stranke podjetja

Funkcionalnost s pogledom na stike in potencialne stranke podjetja je implementirana s pogledom vseh stikov in potencialnih strank podjetja z možnostjo preproste označitve, ali je izbran kontakt trenutno potencialna stranka podjetja. Zraven imena, priimka in pozicije so prikazani še prvi e-poštni naslov stika, prva telefonska številka stika ter podjetje, v katerem je kontakt zaposlen. Novi stiki in potencialne stranke se dodajajo preko fiksno pozicioniranega gumba. Delovanje funkcionalnosti je ponazorjeno s Petrijevo mrežo na sliki 3.3.

3.4.4 Funkcionalnost pregleda aktivnosti in dogodkov

Funkcionalnost pregleda aktivnosti in dogodkov je implementirana s pogledom na trenutni teden. V pogledu so na uro natančno prikazani vsi dogodki in naloge, ki jih ima uporabnik vnešene v trenutnem tednu.

3.4.5 Funkcionalnost vpogleda statistik uporabe sistema

Funkcionalnost vpogleda statistik vsebuje določene statistike, ki se beležijo ob uporabi sistema. Te vključujejo sledeče podatke:

- število ustvarjenih poslov v določenem časovnem obdobju,
- število dobljenih in izgubljenih poslov v določenem časovnem obdobju,

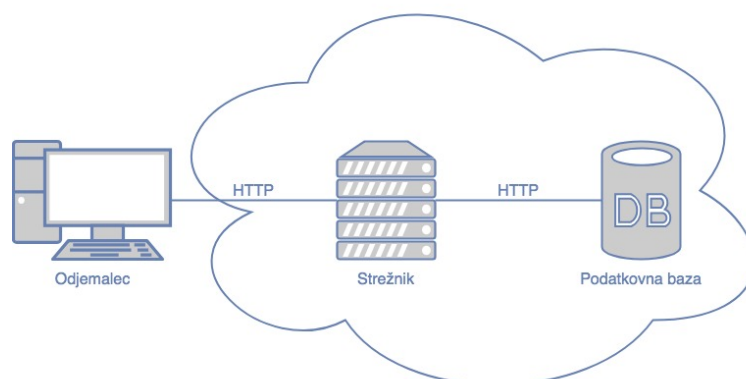
- število ustvarjenih opravil v določenem časovnem obdobju,
- število opravljenih opravil v določenem časovnem obdobju,
- število ustvarjenih stikov v določenem časovnem obdobju,
- število ustvarjenih potencialnih strank v določenem časovnem obdobju.

4 Izdelava prototipa

Z uspešno definiranimi ključnimi funkcionalnostmi smo se osredotočili na razvoj prototipa oblačnega sistema za podporo delovanja zagonskih podjetij. V pričujočem poglavju je najprej predstavljena zasnova celotnega oblačnega sistema in njegovih struktur. Temu sledi predstavitev ključnih tehnologij izbranih za razvoj prototipa skupaj z razlogi za njihovo izbiro. Po predstavitvi sledi podrobno prikazan postopek razvoja in delovanja struktur predstavljenih v razdelku 4.1. Po predstavitvi struktur sledi še pregled delovanja posameznih gradnikov prototipa, ki implementirajo ključne funkcionalnosti predstavljene v razdelku 3.4.

4.1 Arhitektura informacijskega sistema

Zaradi izbire razvoja oblačnega sistema je bila pri pristopu zasnove prototipa ključna izbira široko podprtih spletnih tehnologij, ki so kompatibilne z večino modernih sistemov ter ozir na možnost nadaljnega razvoja oblačnega sistema. Izhodišče zasnove je bila odločitev, da bodo uporabniki oblačni sistem uporabljali s pomočjo enega izmed modernih spletnih brskalnikov. V njem bodo uporabljali spletno aplikacijo, ki bo po principu



Slika 4.1 Zasnova oblachnega sistema z njegovimi strukturami.

odjemalec-strežnik na strežniški sistem na osnovi protokola HTTP pošiljala zahteve. Posledično se mora zaradi tega na strežniškem sistemu izvajati strežniška aplikacija, ki odgovarja na odjemalčeve zahteve. Končno je bilo potrebno izbrati tudi način shranjevanja podatkov, ki se generirajo skozi uporabo spletne aplikacije oblachnega sistema. Za shranjevanje le-teh smo izbrali podatkovno bazo, pri čemer mora biti ta zmožna komunikacije s strežniško aplikacijo na osnovi HTTP protokola. Končen pregled in povezava omenjenih struktur takšnega oblachnega sistema je prikazana na sliki 4.1.

4.2 Uporabljene tehnologije

Pri izbiri tehnologij za razvoj oblachnega sistema predstavljenega v podpoglavju 4.1 smo izhajali iz programskih jezikov, ki jih nujno potrebujemo za razvoj moderne spletne aplikacije. Ti vključujejo programske jezike HTML, CSS in Javascript. Izmed jezikov je ključen predvsem Javascript, s katerim je implementirana logika delovanja spletne aplikacije. V oziru na to smo iskali kombinacijo rešitev, s katerimi bi lahko na njegovi osnovi razvili logiko delovanja celotnega oblachnega sistema. Ta vključuje:

- delovanje zmogljive spletne aplikacije,
- delovanje strežniške aplikacije,
- implementacijo podatkovne baze,
- povezavo strežniške aplikacije s podatkovno bazo.

Naša končna izbira tehnologij, s katerimi je bila implementirana logika delovanja celotnega oblačnega sistema, je denominirana s kratico *MERN*. Omenjene rešitve vključujejo:

- MongoDB (uporabljen za podatkovno bazo),
- Express.js (uporabljen za razvoj strežniške aplikacije),
- React.js (uporabljen za razvoj spletne aplikacije),
- Node.js (uporabljen za izvajalno okolje strežniške aplikacije ter pri razvoju).

V nadaljevanju sledi predstavitev omenjenih tehnologij ter razlogi za izbiro pri razvoju prototipa oblačnega sistema.

4.2.1 MongoDB

MongoDB je dokumentna podatkovna baza s skalabilnostjo in fleksibilnostjo, ki se uporablja za iskanje in indeksiranje podatkov [30]. MongoDB je podatkovna baza tipa NoSQL, pri čemer so podatki v MongoDB podatkovni bazi shranjeni v obliki podobni JSON notaciji. Prednost tega je, da se lahko skozi dokumentni podatkovni model lahko dodaja vnose s poljubno strukturo in da se lahko dinamično modificira shema posameznega modela. Zaradi različnih zahtev števila zagonskih podjetij je to idealna izbira za dodatne prilagoditve sistema, v kolikor bi jih zagonsko podjetje potrebovalo. Ključno je tudi, da je MongoDB odprtokodna rešitev, ki je podprta pri številu ponudnikov gostovanja podatkovnih baz brez dodatnih nastavitvev.

4.2.2 Express.js

Express.js je minimalistično in fleksibilno Node.js strežniško aplikacijsko ogrodje, ki ponuja robustni nabor funkcionalnosti za spletne in mobilne aplikacije [31]. Na podlagi številnih implementiranih metod, ki omogočajo REST zahteve, lahko na podlagi ogrodja Express.js preprosto razvijemo strežniško aplikacijo, ki je povezana s podatkovno bazo. V našem primeru je primaren razlog izbire dejstvo, da oblačni sistem ne potrebuje kompleksne strežniške aplikacije, saj le-ta servira enostransko spletno aplikacijo (angl. *Single Page Application*) [36]. Dodatno se lahko pri nadaljnjem razvoju na osnovi Express.js preprosto dodajo nove funkcionalnosti API-ja.

4.2.3 React.js

React.js je odprtokodna knjižnica, ki omogoča razvoj modernih uporabniških vmesnikov [32]. Knjižnica je bila razvita pri podjetju Facebook kot odgovor na kompleksnost razvoja njihovih rešitev. Izbira knjižnice React.js pri razvoju oblačnega sistema je bazirana na implementaciji dveh naprednih tehnologij v knjižnici. Prva izmed teh je Virtual DOM, ki omogoča, da nam ob vsaki spremembi ali zahtevi ni potrebno ponovno naložiti celotne spletne aplikacije, temveč samo del spletne aplikacije, ki se dejansko spremeni. Konkretno Virtual DOM v spominu vedno drži trenutno kopijo HTML DOM-a in vse spremembe opravlja nad to kopijo. Nato to kopijo primerja z dejanskim HTML DOM-om in v njem zamenja samo nujno potrebne dele. Zaradi tega so spletne aplikacije, ki implementirajo to tehnologijo, v primerjavi z ostalimi spletnimi aplikacijami mnogo bolj zmogljive. Drug razlog je, da knjižnica pri razvoju uporablja JSX (razširitev programskega jezika Javascript). Ta pri razvoju omogoča, da so posamezni deli spletne aplikacije lahko preprosto razdeljeni na vedno manjše komponente, kar prinaša boljšo berljivost ter strukturiranost programske kode. Nadalje, dobra zasnova teh komponent je ključna pri učinkovitosti Virtual DOM-a.

4.2.4 Node.js

Node.js je odprtokodno izvajalno okolje, ki omogoča izvajanje programske kode razvite v programskem jeziku Javascript [33],[34]. Čeprav je bila tehnologija Node.js zasnovana za uporabo na strežniških sistemih se danes vedno pogosteje uporablja pri razvoju spletnih ter strežniških aplikacij. V našem primeru Node.js izvajalno okolje nujno potrebujeta tehnologiji Express.js in React.js. Pri tem ga React.js potrebuje izključno pri razvoju, Express.js pa pri razvoju ter izvajanju aplikacije, ki je zgrajena na njegovi osnovi.

4.3 Predstavitev struktur oblačnega sistema

Kot je razvidno v razdelku 4.1, so strukture oblačnega sistema spletna aplikacija, strežniška aplikacija ter podatkovna baza. V nadaljevanju sledi podrobna predstavitev vseh omenjenih struktur skupaj z načinom njihove implementacije ter izvedbe razvoja.

4.3.1 Predstavitev spletne aplikacije

Začetek razvoja spletne aplikacije sloni na sprejeti odločitvi, da se bo ta lahko izvajala v kateremkoli izmed modernih spletnih brskalnikov. Ker je ključna uporabljena tehnologija spletne aplikacije knjižnica React.js, morajo ti brskalniki podpirati različice tehnologij HTML 5, CSS3 in Ecmascript 5 za pravilno delovanje. Na osnovi tehnologije Virtual DOM [35], ki je del knjižnice React.js, je spletna aplikacija razvita kot enostranska aplikacija. Posledično spletna aplikacija vse potrebne datoteke za delovanje zahteva od strežniškega sistema izključno ob njenem prvotnem nalaganju. Po prvotnem nalaganju spletna aplikacija na strežniški sistem pošilja izključno le še API zahteve, s katerimi pridobiva podatke iz podatkovne baze. Nadalje nam tehnologija Virtual DOM omogoča, da ob zahtevku po novi strani spletne aplikacije spremenimo samo nujno potreben del HTML DOM-a. Zaradi tega potrebuje spletna aplikacija mnogo manj resursov kot bi jih, če bi ob vsakem zahtevku po novi strani morala ponovno spreminjati celoten HTML DOM. Posledično je zaželeno, da se vedno modificira čim manjši del HTML DOM-a. Za doseganje tega so posamezni deli spletne aplikacije od zgoraj navzdol razdeljeni na komponente pri čemer vsaka komponenta vsebuje vedno manjše komponente. V nadaljevanju sledi predstavitev delovanja dveh ključnih komponent spletne aplikacije poimenovani index.js in App.js na osnovi katerih delujejo vse ostale komponente, ki implementirajo gradnike sistema. Delovanje komponent gradnikov je pri tem predstavljeno v razdelku 4.4.

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, browserHistory, IndexRoute } from 'react-router';
import './index.css';
import App from './App';
import PipelineContainer from './containers/pipeline/PipelineContainer';
import TasksContainer from './containers/tasks/TasksContainer';
import ContactsContainer from './containers/contacts/ContactsContainer';
import CalendarContainer from './containers/calendar/CalendarContainer';
import ContactContainer from './containers/contact/ContactContainer';
import StatisticsContainer from './containers/statistics/
  StatisticsContainer';
```

```
ReactDOM.render(  
  <Router history={browserHistory}>  
    <Route path="/" component={App}>  
      <IndexRoute component={PipelineContainer}/>  
      <Route path="/tasks" component={TasksContainer}/>  
      <Route path="/contacts" component={ContactsContainer}/>  
      <Route path="/contact/:userId" component={ContactContainer}/>  
      <Route path="/calendar" component={CalendarContainer} />  
      <Route path="/statistics" component={StatisticsContainer} />  
    </Route>  
  </Router>,  
  document.getElementById('root')  
)  
);
```

Programska koda 4.1 index.js datoteka

Komponenta v okviru datoteke index.js, ki je prikazana v razseku 4.1, je najosnovnejša zahtevana datoteka v okviru React.js knjižnice. Ta v spletni aplikaciji implementira logično najvišje pozicionirano komponento, ki svojo vsebino dodaja v element s ključem root v datoteki index.html. V spletni aplikaciji je poleg omenjenega njena naloga tudi, da se na podlagi tehnologije React Router [37] odziva na spremembe URI naslova v brskalniku in ob tem dogodku komponenti app.js posreduje ustrezno komponento za izris. Datoteka index.js se v okviru spletne aplikacije odziva na sledeče URI spremembe:

- / posreduje komponento pipelineContainer.js,
- /tasks posreduje komponento tasksContainer.js,
- /contacts posreduje komponento contactsContainer.js,
- /contact/:userId posreduje komponento contactContainer.js,
- /calendar posreduje komponento calendarContainer.js,
- /statistics posreduje komponento statisticsContainer.js.

```
import React, { Component } from 'react';  
import './css/app.css';  
import NavbarContainer from './containers/NavbarContainer';  
import AddFunctionContainer from './containers/modals/AddFunctionContainer';
```

```
class App extends Component {  
  
  constructor(props, context){  
    super(props, context);  
  
    this.setState = this.setState.bind(this);  
  }  
  
  render() {  
    return (  
      <div>  
        <NavbarContainer/>  
        <section className="outerContainer">  
          {this.props.children}  
          <AddFunctionContainer/>  
        </section>  
      </div>  
    );  
  }  
}  
  
export default App;
```

Programska koda 4.2 App.js datoteka

Programska koda v okviru datoteke App.js, ki je prikazana v razseku 4.2, je primarno odgovorna za izrisovanje komponent, ki jih dobi posredovane preko index.js komponente. Prav tako izrisuje komponento NavbarContainer, ki implementira navigacijsko vrstico spletne aplikacije ter komponento AddFunctionContainer, ki implementira vedno dostopne načine dodajanja poslov, nalog ali stikov v okviru oblačnega sistema.

4.3.2 Predstavitev strežniške aplikacije

Strežniška aplikacija je implementirana z uporabo tehnologije Express.js. Ta nam omogoča, da smo v okviru datoteke server.js, ki je prikazana v programski kodi 4.3, izdelali strežniško aplikacijo, ki na zahteve posluša na vtičniku 4000. Za nazornost programske kode je pri tem izpuščena logika delovanja API odzivnih točk. Strežniška aplikacija v okviru oblačnega sistema opravlja dve nalogi. Prva je, da servira spletno aplikacijo oblačnega sistema in njene statične vsebine. Pri tem je serviranje spletne aplikacije zelo preprosto, saj je spletna aplikacija enostranska aplikacija. Druga je, da implementira

RESTful API za pridobivanje ter manipulacijo s podatki iz podatkovne baze. Osnova našega RESTful API-ja je vtičnik MongoClient, ki nam omogoča povezavo z našo MongoDB podatkovno bazo. Z njim pridobivamo ter manipuliramo s podatki iz naše podatkovne baze. Na podlagi tega smo po principu REST razvili API odzivne točke, ki podpirajo eno ali več izmed GET, PUT, POST ali DELETE zahtev za posamezen podatkovni tip. Posamezne API odzivne točke so opisane pri vsakem posameznem gradniku v razdelku 4.4, pri čemer RESTful API strežniške aplikacije omogoča akcije nad sledečimi objekti:

- posli,
- nalogami,
- stiki,
- aktivnostmi,
- statistikami.

```
const express = require('express');
const bodyParser = require('body-parser');
const MongoClient = require('mongodb').MongoClient;
const mongoose = require('mongoose');
const ObjectId = require('mongodb').ObjectId;
const static_path = __dirname + '/build/';
const server = express();
const db_path = 'mongodb://admin:<geslo>@localhost:27017/xCRM?authMechanism=DEFAULT&authSource=admin';
let db;

// Povezava z MongoDB podatkovno bazo
MongoClient.connect(db_path, (err, res) => {...});

// Middleware
server.use(express.static(static_path));
server.use(bodyParser.json());
server.use(bodyParser.urlencoded({extended: true}));
server.use((req, res, next) => {
  res.header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE');
  res.header('Access-Control-Allow-Headers', 'Content-Type');
  next();
});
```

```
});

// GET API odzivne tocke
server.get('/api/pipeline', (req, res) => {...});
server.get('/api/pipeline/:pipelineId/stage/:stageId', (req, res) => {...})
;
server.get('/api/deals/:query=:value', (req, res) => {...});
server.get('/api/contacts', (req, res) => {...});
server.get('/api/contact/:userId', (req, res) => {...});
server.get('/api/tasks/:query=:value', (req, res) => {...});
server.get('/api/calendar/:startYear-:startMonth-:startDay-:endYear-:
    endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/deals/created/:startYear-:startMonth-:startDay
    -:endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/deals/won/:startYear-:startMonth-:startDay-:
    endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/deals/lost/:startYear-:startMonth-:startDay-:
    endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/tasks/created/:startYear-:startMonth-:startDay
    -:endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/tasks/finished/:startYear-:startMonth-:startDay
    -:endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/contacts/created/:startYear-:startMonth-:
    startDay-:endYear-:endMonth-:endDay', (req, res) => {...});
server.get('/api/statistics/contacts/leads/:startYear-:startMonth-:startDay
    -:en', (req, res) => {...});

// POST API odzivne tocke
server.post('/api/deal', (req, res) => {...});
server.post('/api/task', (req, res) => {...});
server.post('/api/contact', (req, res) => {...});

// PUT API odzivne tocke
server.put('/api/contact', (req, res) => {...});
server.put('/api/deal', (req, res) => {...});
server.put('/api/task', (req, res) => {...});

// DELETE API odzivne tocke
server.delete('/api/contact', (req, res) => {...});

// Serviranje spletne aplikacije
```

```
server.get('*', (req, res) => {
  res.sendFile(static_path+"index.html");
});

// Zagon strezniške aplikacije
server.listen(4000, () => {
  console.log('Server is listening on port 4000');
});
```

Programska koda 4.3 server.js datoteka

4.3.3 Predstavitev implementacije podatkovne baze

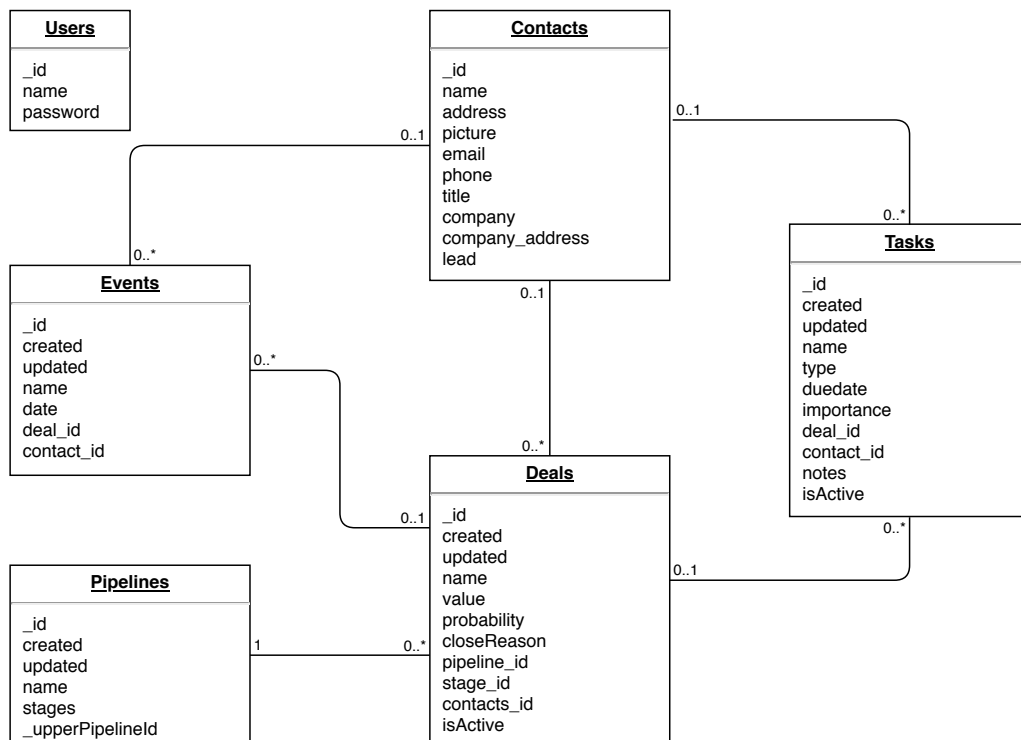
Zadnji gradnik sistema je podatkovna baza implementirana s tehnologijo MongoDB. Podatkovna baza MongoDB v okviru enega dokumenta vsebuje zbirke poimenovane "Collection" v katerih so zajeti vnosi, ki vsebujejo različne podatkovne tipe. Shema celotne podatkovne baze s podatkovnimi tipi oblachnega sistema je prikazana na diagramu na sliki 4.2. Pri tem je diagram podatkovne baze modeliran po predlogu iz vira [38]. V nadaljevanju sledi opis glavnih podatkovnih tipov posameznih "Collection" zbirk.

Collection Users vsebuje uporabniško ime posameznega uporabnika podatkovne baze in njegovo geslo. Zbirka se uporablja pri avtentikaciji s podatkovno bazo.

Collection Pipelines vsebuje posamezne cevovode z njihovim imenom ter njegovimi posameznimi stopnjami. Cevovodi lahko vsebujejo tudi referenco na drug cevovod, v kolikor mu je sam podrejen.

Collection Deals vsebuje podatke posameznih poslov v oblacnem sistemu (ime, vrednost, verjetnost in metapodatke), referenco na cevovod, v katerem se nahaja, ime stopnje, v kateri se nahaja in označitev, ali je posel še aktiven. Nadalje lahko posamezen posel vsebuje tudi referenco na posamezen stik in razloge pri zaključku posla.

Collection Contacts vsebuje podatke posameznih stikov v oblacnem sistemu (ime, naslov stika, e-poštni naslov stika, telefonsko številko stika in metapodatke), njegove podatke v podjetju za katerega dela (ime podjetja, naslov podjetja in njegovo pozicijo v njem) in ali je stik poslovna tarča.



Slika 4.2 Diagram podatkovne baze.

Collection Tasks vsebuje podatke posamezne naloge v oblacnem sistemu (ime, vrsto, rok, pomembnost in metapodatke) in ali je naloga še aktivna. Nadalje lahko posamezna naloga vsebuje tudi referenco na posamezen posel in referenco na posamezen stik.

Collection Events vsebuje podatke posameznega dogodka v oblacnem sistemu (ime, datum in metapodatke). Dogodek lahko vsebuje tudi referenco na posamezen posel in referenco na posamezen stik.

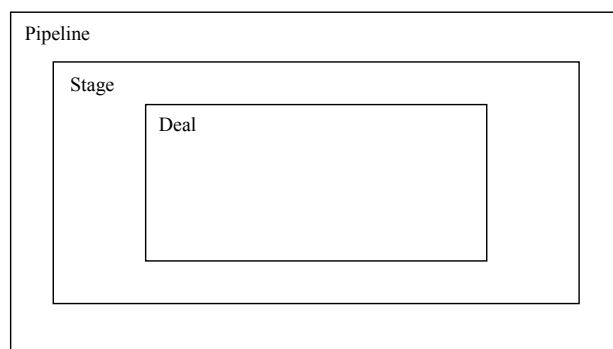
4.4 Delovanje prototipa

V pričujočem razdelku se fokusiramo na delovanje gradnikov, ki implementirajo posamezne ključne funkcionalnosti oblacnega sistema. V nadaljevanju sledi pregled njihove zasnove in izvedba njihovega delovanja.

4.4.1 Delovanje cevovodnega pogleda

Gradnik cevovodnega pogleda je pri spletni aplikaciji logično sestavljen iz komponent cevovoda (pipeline), stopnje (stage) in posla (deal). Pri tem lahko cevovod vsebuje eno ali več stopenj in vsaka izmed stopenj lahko vsebuje enega ali več poslov. Hierarhija omenjenih komponent cevovodnega pogleda je prikazana na sliki 4.3. Za izvršitev akcij pri gradniku cevovodnega pogleda, ki so predstavljene na sliki 3.1, so v okviru API strežniške aplikacije pripravljene ustrezne odzivne točke. Te so:

- pridobivanje vseh cevovodov, stopenj in poslov z GET zahtevkom,
- posodobitev pri premiku posla iz ene stopnje v drugo s PUT zahtevkom,
- posodobitev pri označitvi posla kot dobljenega ali izgubljenega s PUT zahtevkom.

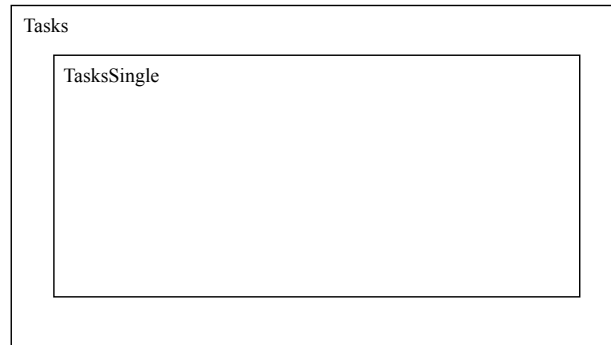


Slika 4.3 Hierarhija komponent cevovodnega pogleda.

4.4.2 Delovanje opravil

Gradnik z opravili je pri spletni aplikaciji logično sestavljen iz okvirne komponente Tasks, ki vsebuje eno ali več posameznih nalog (TasksSingle). Hierarhija omenjenih komponent pri gradniku opravil je pri tem prikazana na sliki 4.4. Za izvršitev akcij pri gradniku opravil, ki so predstavljene na sliki 3.2, so v okviru API strežniške aplikacije pripravljene ustrezne odzivne točke. Te so:

- pridobivanje vseh aktivnih opravil z GET zahtevkom,
- posodobitev podatkov posamezne naloge s PUT zahtevkom,
- posodobitev pri označitvi naloge kot končane s PUT zahtevkom.



Slika 4.4 Hierarhija komponent opravil.

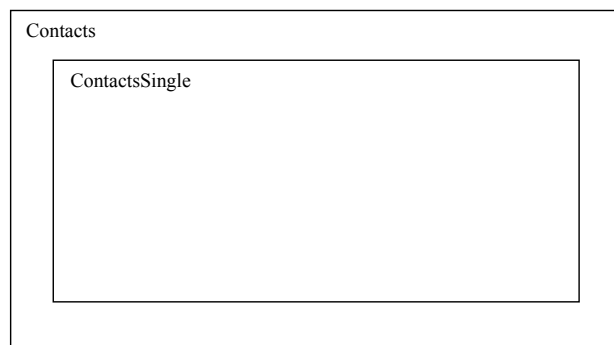
4.4.3 Delovanje upravljanja s stiki in potencialnimi strankami

Gradnik upravljanja s stiki in potencialnimi strankami je pri spletni aplikaciji logično sestavljen iz okvirne komponente (Contacts), ki vsebuje komponento enega ali več izmed posameznih stikov (ContactsSingle). Hierarhija delovanja omenjenih komponent je prikazana na sliki 4.5. Dodatno je pri gradniku možen vpogled nad podatki in aktivnostmi nad posameznim stikom. Hierarhija delovanja vpogleda je prav tako implementirana s komponentami in je prikazana na sliki 4.6. Ta vsebuje okvirno komponento (Contact), ki vsebuje komponento s podatki stika (ContactDetail) in komponento z aktivnostmi pri posameznim stikom (ContactActivity). Za izvršitev akcij pri gradniku upravljanja s stiki in potencialnimi strankami, ki so predstavljene na sliki 3.3, so v okviru API strežniške aplikacije pripravljene ustrezne odzivne točke. Te so:

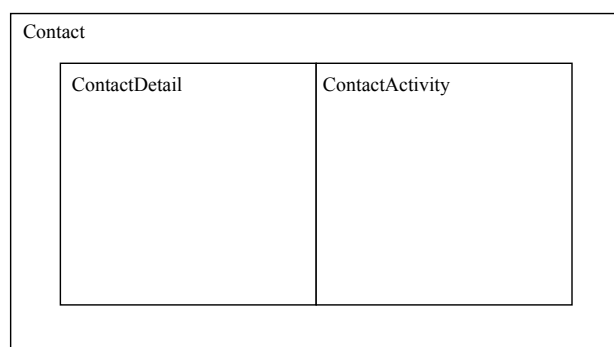
- pridobivanje vseh stikov z GET zahtevkom,
- posodobitev pri označitvi stika kot stranko z PUT zahtevkom,
- izbris stika z DELETE zahtevkom.

4.4.4 Delovanje pogleda aktivnosti in dogodkov z opravili

Gradnik pogleda aktivnosti in dogodkov z opravili je pri spletni aplikaciji logično sestavljen iz okvirne komponente (Calendar), komponente, ki predstavlja dan (CalendarDay) in komponente, ki vsebuje eno uro posameznega dneva (CalendarHour). Pri tem vsebuje komponenta Calendar sedem komponent CalendarDay, ena izmed komponent CalendarDay pa vsebuje štiriindvajset komponent CalendarHour. Hierarhija omenjenih

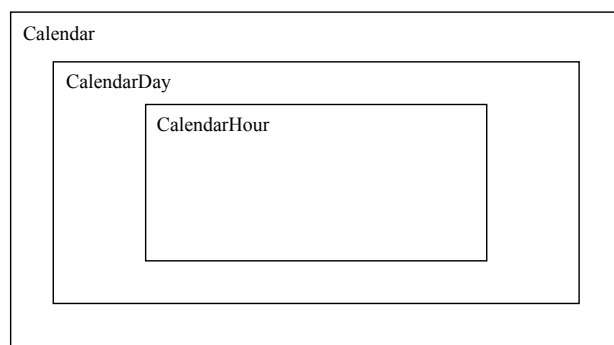


Slika 4.5 Hierarhija komponent upravljanja s stiki in strankami.



Slika 4.6 Hierarhija komponent posameznega stika.

komponent je prikazana na sliki 4.7. Za izvršitev akcij pri gradniku vpogleda aktivnosti in dogodkov z opravili je v okviru API vmesnika na strežniški aplikaciji pripravljena ena ustrezna odzivna točka. Ta je pridobivanje vseh opravil in aktivnosti z GET zahtevkom za izbran teden.



Slika 4.7 Hierarhija pogleda aktivnosti ter dogodkov z opravili.

4.4.5 Delovanje vpogleda statistik

Gradnik pogleda statistik je logično sestavljen izključno iz okvirne komponente (Statistics). Za izvršitev akcij pri gradniku so v okviru API strežniške aplikacije pripravljene ustrezne odzivne točke. Te so:

- pridobivanje števila vseh ustvarjenih poslov v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh dobljenih poslov v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh izgubljenih poslov v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh ustvarjenih nalog v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh opravljenih nalog v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh ustvarjenih stikov v določenem časovnem obdobju z GET zahtevkom,
- pridobivanje števila vseh ustvarjenih potencialnih strank v določenem časovnem obdobju z GET zahtevkom.

4.5 Vzpostavitev oblachnega sistema

S koncem razvoja sistema je bilo potrebno sam sistem tudi vzpostaviti za namene dostopa preko spleta. Pri tem smo imeli pri sami vzpostavitvi številne možnosti pri izbiri ponudnikov gostovanja tako za spletno ter strežniško aplikacijo kot tudi za gostovanje podatkovne baze. Ker je ena izmed ključnih lastnosti oblachnega sistema njegova odprtost smo iskali rešitve, s katerimi lahko brezplačno vzpostavimo celoten oblachni sistem. V našem primeru smo za brezplačno gostovanje spletne ter strežniške aplikacije izbrali ponudnika Heroku, za gostovanje podatkovne baze pa rešitev ponudnika MongoDB Atlas. V nadaljevanju sledijo razlogi izbire obeh ponudnikov, proces vzpostavitve delov oblachnega sistema pri obeh ponudnikih in nekaj besed o samih zahtevah oblachnega sistema.

4.5.1 Vzpostavitev spletne in strežniške aplikacije

Kot omenjeno smo za vzpostavitev spletne in strežniške aplikacije izbrali ponudnika Heroku. Pri tem naša odločitev poleg možnosti brezplačnega gostovanja sloni na dejstvu, da nam Heroku privzeto brez dodatnih potrebnih konfiguracij omogoča vzpostavitev Node.js izvajalnega okolja skupaj z git repozitorijem za namene razvoja in nalaganja spletne ter strežniške aplikacije na njihove sisteme. Nadalje nam Heroku ob vzpostavitvi samodejno pripravi poddomeno, preko katere lahko s poljubnim brskalnikom dostopamo do spletne aplikacije. Pri tem so zaslonski izgledi omenjene spletne aplikacije prikazani v prilogi A. Proces vzpostavitve spletne in strežniške aplikacije je bil v našem primeru sledeč:

1. Pri ponudniku Heroku registriramo uporabniški račun.
2. Za stvaritev novega projekta namestimo Heroku CLI.
3. Za naš projekt ustvarimo novo mapo, se premaknemo vanjo ter v njej poženemo ukaz *heroku create*.
4. V mapo projekta skopiramo datoteko *server.js* in produkcijske datoteke spletne aplikacije.
5. V mapi poženemo ukaz *npm init* in izpolnimo zahtevane podatke.
6. V novo ustvarjeni datoteki *package.json* v delu *scripts* dodamo "start": "server.js".
7. S sosledjem ukazov *git add*, *git commit* in *git push heroku master* na Heroku naložimo spletno ter strežniško aplikacijo.
8. Po uspešno izvedenih ukazih nam Heroku samodejno pripravi celoten URL naslov, preko katerega lahko dostopamo do spletne aplikacije.

4.5.2 Vzpostavitev podatkovne baze

Za vzpostavitev naše podatkovne baze MongoDB smo izbrali ponudnika MongoDB Atlas. Izbira ponudnika pri tem sloni na dejstvu, da je rešitev MongoDB Atlas rešitev podjetja, ki prav tako razvija samo tehnologijo MongoDB. Z njihovo rešitvijo MongoDB Atlas lahko tako hitro in samodejno vzpostavimo novo MongoDB podatkovno bazo. Nadalje nam MongoDB Atlas omogoča izbiro, ali želimo, da naša podatkovna baza gostuje v okviru sistema Amazon AWS, sistema Google Cloud Platform ali sistema Microsoft Azure. V

našem primeru smo se odločili za gostovanje pri Google Cloud Platform, saj ta ponuja možnost brezplačnega gostovanja. Proces vzpostavitve podatkovne baze je bil v našem primeru tako sledeč:

1. Pri ponudniku MongoDB Atlas registriramo uporabniški račun.
2. V okviru njihove rešitve ustvarimo nov projekt in v njem zaženemo novo gručo.
3. V namestitvenem meniju izberemo ponudnika Google Cloud Platform, eno izmed brezplačnih lokacij za gostovanje ter gručo tipa M0.
4. V namestitvenem meniju po želji poimenujemo našo gručo.
5. V okviru gruče ustvarimo novega uporabnika z administratorskimi pravicami.
6. V okviru gruče skopiramo generiran URI naslov in ga dodamo na ustrezno mesto v datoteki server.js. Pri tem v URI naslovu del <PASSWORD> zamenjamo z geslom uporabnika z administratorskimi pravicami.

4.5.3 Zahteve oblačnega sistema

Vsaka posamezna struktura oblačnega sistema je v oziru na zmogljivost modernih sistemov nezahtevna. Spletna aplikacija se lahko brez težav izvaja na vseh sistemih, ki podpirajo enega izmed modernih brskalnikov. Strežniška aplikacija se lahko brez težav izvaja tudi na sistemih kot so Raspberry Pi, dokler ti podpirajo Node.js izvajalno okolje, sama podatkovna baza pa je prostorsko nezahtevna v kolikor stiki ne vsebujejo slik. Zagonska podjetja bi lahko tako celoten oblačni sistem brez problema vzpostavila tudi na njihovih internih strežnikih, v kolikor bi to želela. V primeru naše vzpostavitve je ozko grlo zagotovo podatkovna baza, saj je ta prostorsko omejena na 512MB. Po naših testiranjih z različnimi količinami in vrstami vnosov ocenjujemo, da lahko podatkovna baza vsebuje okoli dva milijona unikatnih vnosov v kolikor stiki ne vsebujejo slik.

5 Sklepne ugotovitve in zaključek

Ozirajoč se na izvedeno raziskovanje informacijskih potreb zagonskih podjetij smo le te uspešno definirali. Slednje omogočajo samo delovanje zagonskih podjetij od ustanovitve naprej, proces pridobivanja investicij in proces izbire informacijskih sistemov.

Na podlagi teh ugotovitev smo uspešno definirali in overili ključne funkcionalnosti sistema za vodenje odnosov s strankami in pridobivanjem alternativnih virov financiranja zagonskih podjetij. Te smo definirali na podlagi analize obstoječih sistemov na tržišču, intervjujev s fokusno skupino angelskih investorjev in intervjuji z zagonskimi podjetji.

Na podlagi definiranja ključnih funkcionalnosti smo uspešno razvili odprtokodni oblačni prototip takšnega sistema. Pri tem smo sistem zasnovali tako, da lahko poljubno zagonsko podjetje z manjšimi spremembami v programski kodi sistem prilagodi zase oziroma ga nadgradi za svoje potrebe. S tem smo uspešno opravil vse zastavljene cilje diplomskega dela.

Kljub temu ima prototip sistema omejitve pri uporabi in potencial za izboljšanje s prilagoditvami. Primarno bi se prototip sistema lahko dodatno nadgradil z implementacijo knjižnice Electron [39]. Z njo bi se lahko sistem uporabljal tudi lokalno na sistemih z operacijskimi sistemi Windows, MacOS in Linux. Čeprav se uporaba sistema ne bi spremenila (v primerjavi s tisto preko spletnega brskalnika), bi bilo smiselno, da lahko uporabniki izbirajo, na katerem operacijskem sistemu želijo uporabljati rešitev. Nekatere druge možne izboljšave in prilagoditve vključujejo integracije s sistemi kot so Google Stiki ali Mailchimp in možnost dodajanja različnih statistik direktno v uporabniškem vmesniku. Primarna omejitev prototipa sistema je, da je modifikacije nastavitev potrebno spremeniti neposredno v programski kodi, kar lahko za nekatere manj tehnično večje zaposlene v zagonskih podjetjih predstavlja omejitev. Nenazadnje je omejitev tudi, da je v prototipu možno imeti samo en tip uporabnika, kar pomeni, da lahko vsi uporabniki sistema opravljajo iste funkcije.

LITERATURA

- [1] Sanford L. Moskowitz. Chapter 2 - The startup and spinoff firm. In *Cybercrime and Business*, pages 25 – 43. Butterworth-Heinemann, Boston, 2017.
- [2] Steve Blank and Bob Dorf. *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K and S Ranch Inc., 2012.
- [3] Carlos Devece, Marta Peris-Ortiz, and Carlos Rueda-Armengot. Entrepreneurship during economic crisis: Success factors and paths to failure. *Journal of Business Research*, 69(11):5366 – 5370, 2016.
- [4] Jeannette Paschen. Choose wisely: Crowdfunding through the stages of the startup life cycle. *Business Horizons*, 60(2):179 – 188, 2017.
- [5] Tahereh Hasani, Jamil Bojei, and Ali Dehghantanha. Investigating the antecedents to the adoption of SCRM technologies by start-up companies. *Telematics and Informatics*, 34(5):655 – 675, 2017.
- [6] Steve Blank What's A Startup? First Principles. Dosegljivo: <https://steveblank.com/2010/01/25/whats-a-startup-first-principles/>, 2010. [Dostopano 4. 8. 2017].
- [7] Eric Ries. *The Lean Startup: How constant innovation creates radically successful businesses*. Portfolio-Penguin, 2011.
- [8] Startup. Dosegljivo: <http://www.investopedia.com/terms/s/startup.asp/>, 2017. [Dostopano 4. 8. 2017].
- [9] What The Hell Is A Startup Anyway? — TechCrunch. Dosegljivo: <https://techcrunch.com/2014/12/30/what-the-hell-is-a-startup-anyway/>, 2014. [Dostopano 4. 8. 2017].

- [10] Startup = Growth . Dosegljivo: <http://www.paulgraham.com/growth.html>, 2012. [Dostopano 6. 8. 2017].
- [11] At what point are you not a startup anymore? — Built in Boston. Dosegljivo: <http://www.builtinboston.com/2017/03/01/what-point-are-you-not-startup-anymore>, 2017. [Dostopano 4. 8. 2017].
- [12] Ash Maurya. *Delaj vitko: od načrta A do načrta, ki deluje*. Založba Pasadena d.o.o., 2014.
- [13] Joseph C. Picken. From startup to scalable enterprise: Laying the foundation. *Business Horizons*, 60(5):587 – 595, 2017.
- [14] S.C. Oranburg. 4 - start-up financing. In *Start-Up Creation*, pages 57 – 73. Woodhead Publishing, 2016.
- [15] Startup spending guide: Where to spend money — TechCrunch. Dosegljivo: <https://techcrunch.com/2016/05/20/where-to-spend-money>, 2016. [Dostopano 10. 9. 2017].
- [16] Kevin C. Cox, Jason Lortie, and Steven A. Stewart. When to pray to the angels for funding: The seasonality of angel investing in new ventures. *Journal of Business Venturing Insights*, 7:68 – 76, 2017.
- [17] Raising Capital: Friends, Family and Fools. Dosegljivo: <http://onstartups.com/tabid/3339/bid/216/Raising-Capital-Friends-Family-and-Fools.aspx>, 2006. [Dostopano 10. 9. 2017].
- [18] How to Get Funding From Friends, Family, and Fools - Business Inside. Dosegljivo: <http://www.businessinsider.com/how-to-get-funding-from-friends-family-and-fools-2010-11>, 2010. [Dostopano 10. 9. 2017].
- [19] Venture Capital. Dosegljivo: <http://www.investopedia.com/terms/v/venturecapital.asp>, 2017. [Dostopano 10. 9. 2017].
- [20] Startups Seeking Funding Should Consider Corporate Venture Capital Arms. Dosegljivo: <https://www.forbes.com/sites/allbusiness/2017/02/25/>

- [startups-seeking-funding-should-consider-corporate-venture-capital-arms/#2609a520759f](#), 2017. [Dostopano 10. 9. 2017].
- [21] The Best Software Tools to Run a Startup. Dosegljivo: <https://www.entrepreneur.com/article/233994>, 2014. [Dostopano 10. 9. 2017].
- [22] Must-have Software for the Better Performance of Your Small Business — HuffPost. Dosegljivo: http://www.huffingtonpost.com/allan-smith/musthave-software-for-the_b_9504168.html, 2016. [Dostopano 10. 9. 2017].
- [23] CRM Market Share Report 2017. Dosegljivo: <http://www.crmsearch.com/crm-software-market-share.php>, 2017. [Dostopano 9. 8. 2017].
- [24] Anil S. Jadhav and Rajendra M. Sonar. Evaluating and selecting software packages: A review. *Information and Software Technology*, 51(3):555 – 563, 2009.
- [25] Sales Cloud: Sales Force Automation CRM Software Features - Salesforce.com. Dosegljivo: <https://www.salesforce.com/products/sales-cloud/features/>, 2017. [Dostopano 24. 9. 2017].
- [26] Sales CRM — SAP. Dosegljivo: <https://www.sap.com/products/customer-relationship-management/crm-sales.html>, 2017. [Dostopano 24. 9. 2017].
- [27] Sales Cloud — Customer Experience — Oracle Cloud. Dosegljivo: https://cloud.oracle.com/en_US/opc/sales-cloud/features, 2017. [Dostopano 24. 9. 2017].
- [28] Keep it Simple: The 5 Core Features of an Effective CRM. Dosegljivo: <https://www.salesforce.com/crm/simple/>, 2017. [Dostopano 9. 8. 2017].
- [29] Miha Mraz and Miha Moškon. *Modeliranje računalniških omrežij [na spletu]*. Založba FE in FRI, 2012.
- [30] What is MongoDB? — MongoDB. Dosegljivo: <https://www.mongodb.com/what-is-mongodb>, 2017. [Dostopano 9. 8. 2017].
- [31] Express - Node.js web application framework. Dosegljivo: <https://expressjs.com/>, 2017. [Dostopano 9. 8. 2017].

- [32] React - a Javascript library for building user interfaces. Dosegljivo: <https://facebook.github.io/react/>, 2017. [Dostopano 9. 8. 2017].
- [33] Node.js - Wikipedia. Dosegljivo: <https://en.wikipedia.org/wiki/Node.js>, 2017. [Dostopano 9. 8. 2017].
- [34] Node.js. Dosegljivo: <https://nodejs.org/en/>, 2017. [Dostopano 9. 8. 2017].
- [35] Virtual DOM and Internals - React. Dosegljivo: <https://reactjs.org/docs/faq-internals.html>, 2018. [Dostopano 8. 5. 2018].
- [36] ASP.NET - Single-Page Applications: Build Modern, Responsive Web Apps with ASP.NET. Dosegljivo: <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>, 2013. [Dostopano 8. 5. 2018].
- [37] GitHub - ReactTraining/react-router: Declarative routing for React. Dosegljivo: <https://github.com/ReactTraining/react-router>, 2018. [Dostopano 8. 5. 2018].
- [38] Kwangchul Shin, Chulhyun Hwang, and Hoekyung Jung. NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen's framework. *International Journal of Applied Engineering Research*, 12(5):632 – 636, 2017.
- [39] Electron - Build cross platform desktop apps with JavaScript, HTML, and CSS. Dosegljivo: <https://electron.atom.io/>, 2017. [Dostopano 9. 8. 2017].

A Priloga



A.1 Končen izgled funkcionalnosti

V pričujoči prilogi so predstavljeni zaslonski izgledi vseh implementiranih funkcionalnosti, ki so opisane v razdelku [4.4](#).

A PRILOGA

Pipeline					
Pipeline	Tasks	Contacts	Calendar	Statistics	
Main					
Potential	In talks	Finishing details	Contract signing	Waiting for wire	
Bisha deal Value: 1000\$ Probability: 22%	Bitror deal Value: 1000\$ Probability: 94%	Collaire deal Value: 1000\$ Probability: 46%	Gluid deal Value: 1000\$ Probability: 68%	Oronko deal Value: 1000\$ Probability: 39%	
Kidgrease deal Value: 1000\$ Probability: 34%	Portalis deal Value: 1000\$ Probability: 69%	Delphide deal Value: 1000\$ Probability: 29%	Luxutra deal Value: 1000\$ Probability: 10%	Eweville deal Value: 1000\$ Probability: 55%	
Zilla deal Value: 1000\$ Probability: 66%	Inear deal Value: 1000\$ Probability: 84%	Viocular deal Value: 1000\$ Probability: 92%	Zentix deal Value: 1000\$ Probability: 2%		
Pyramax deal Value: 1000\$ Probability: 41%	Biflex deal Value: 1000\$ Probability: 59%		Mondela deal Value: 10000\$ Probability: 60%		
Bristo deal Value: 1000\$ Probability: 99%			TireForce deal Value: 16000\$ Probability: 60%		
Handshake deal Value: 1000\$ Probability: 77%					
Kampala Deal Value: 170000\$ Probability: 60%					

© 2016 Proton 2016. All rights reserved.



Type	Tasks	Due	Importance	Actions
>	Follow-up Lisa Austin Lisa Austin	Tuesday 5/6/2018	Very low	FOLLOWUP FINISH
☎	Call Charity Mcclure Charity Mcclure	Tuesday 5/6/2018	Very high	FOLLOWUP FINISH
>	Follow-up Elisa Bryan Elisa Bryan	Tuesday 5/6/2018	Low	FOLLOWUP FINISH
☎	Call Owen Ford Owen Ford	Tuesday 5/6/2018	Medium	FOLLOWUP FINISH
☎	Call Violet Stevenson Violet Stevenson	Tuesday 5/6/2018	Medium	FOLLOWUP FINISH
☎	Call Cassandra Carr Cassandra Carr	Tuesday 5/6/2018	Very low	FOLLOWUP FINISH
🗓	Meet with Jeannine Campbell Jeannine Campbell	Tuesday 5/6/2018	High	FOLLOWUP FINISH
>	Follow-up Long Beasley Long Beasley	Friday 8/6/2018	Low	FOLLOWUP FINISH
@	Email Russell Pearson Russell Pearson	Friday 8/6/2018	Very low	FOLLOWUP FINISH
@	Email Martha Ochoa Martha Ochoa	Friday 8/6/2018	Very high	FOLLOWUP FINISH
>	Follow-up Pauline Wallace Pauline Wallace	Sunday 10/6/2018	Medium	FOLLOWUP FINISH
☎	Call Santos Gamble Santos Gamble	Sunday 10/6/2018	Very low	FOLLOWUP FINISH
@	Email Hansen Burke Hansen Burke	Sunday 10/6/2018	Very low	FOLLOWUP FINISH
☎	Call Combs Santiago Combs Santiago	Sunday 10/6/2018	Very low	FOLLOWUP FINISH
☎	Call Moses McIntyre Moses McIntyre	Sunday 10/6/2018	Medium	FOLLOWUP FINISH
@	Email Merisa Snyder Merisa Snyder	Sunday 10/6/2018	Very high	FOLLOWUP FINISH
@	Email Mabel Fulton Mabel Fulton	Sunday 10/6/2018	Very high	FOLLOWUP FINISH
>	Follow-up Florian Mayer Florian Mayer	Tuesday 12/6/2018	Medium	FOLLOWUP FINISH
@	Email Charles Finwatch Charles Finwatch	Wednesday 13/6/2018	Medium	FOLLOWUP FINISH
@	Email Mary Zane Mary Zane	Wednesday 13/6/2018	Medium	FOLLOWUP FINISH

© 2018 Project 5118. All rights reserved.

Slika A.2 Pogled z opravili v sistemu.

A PRILOGA

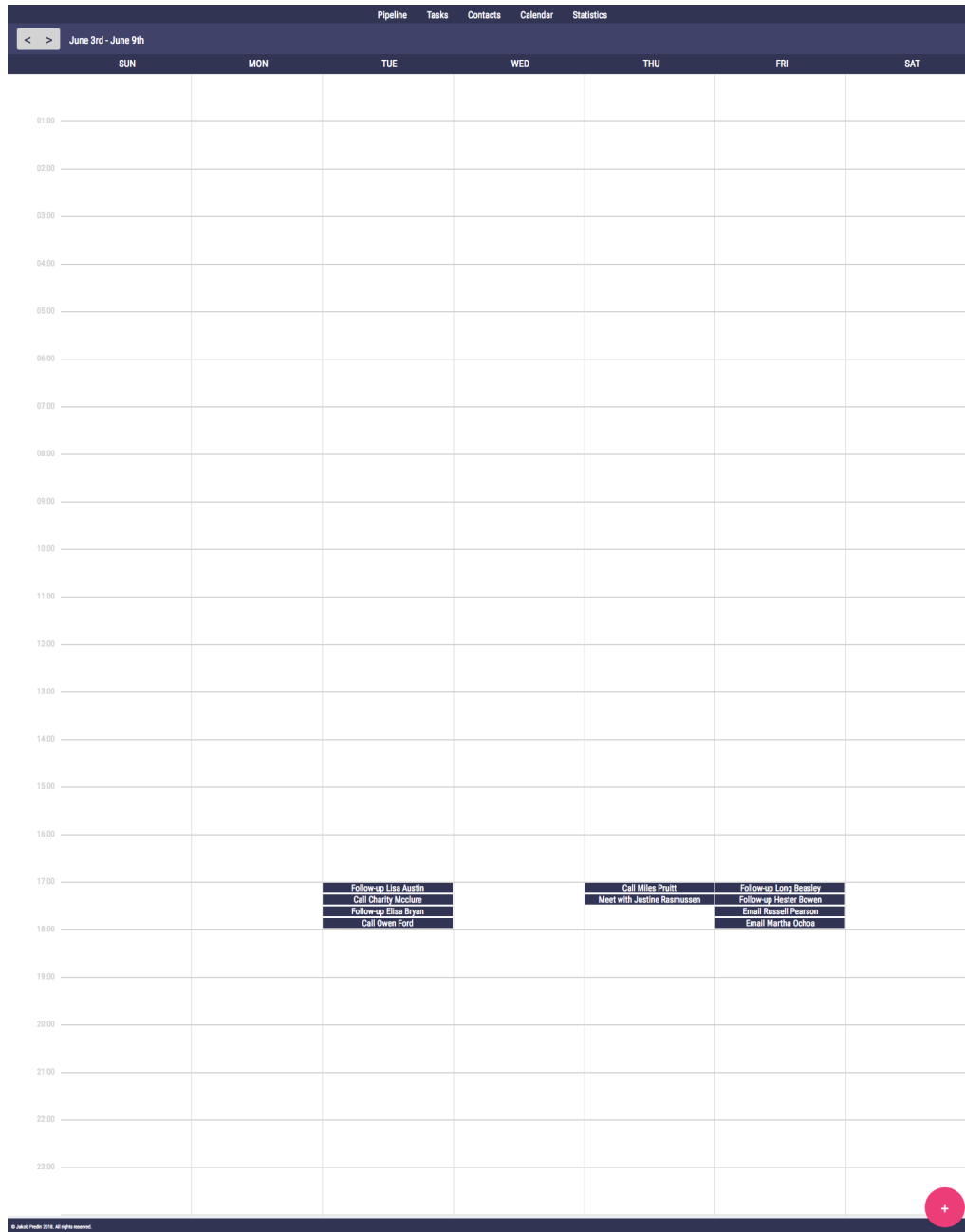
	Name	Email	Tasks	Contacts	Calendar	Statistics	Company	Lead
	George Solomon CEO @ Accidency	georgesolomon@accidency.name			+1 (920) 403-3291		Accidency	<input checked="" type="checkbox"/>
	Milagros English CEO @ Pyramis	milagrosenglish@pyramis.me			+1 (915) 503-3082		Pyramis	<input checked="" type="checkbox"/>
	Danielle Davidson CEO @ Straloy	danielledavidson@straloy.tv			+1 (862) 478-2207		Straloy	<input type="checkbox"/>
	Sosa Delacruz CEO @ Quarx	sosadelacruz@quarx.io			+1 (930) 509-3850		Quarx	<input checked="" type="checkbox"/>
	Mueller Webb CEO @ Sceenty	muellerwebb@sceenty.com			+1 (916) 579-3642		Sceenty	<input type="checkbox"/>
	Adelia Spence CEO @ Comvoy	adeliaspence@comvoy.ca			+1 (877) 416-2843		Comvoy	<input type="checkbox"/>
	Tasha Blanchard CEO @ Aquafire	tashablanchard@aquafire.biz			+1 (905) 597-3680		Aquafire	<input type="checkbox"/>
	Ana Rodriguez CEO @ Nurail	anarodriguez@nurail.net			+1 (930) 555-2909		Nurail	<input checked="" type="checkbox"/>
	Strickland Henson CEO @ Eternis	stricklandhenson@eternis.co.uk			+1 (904) 492-3854		Eternis	<input type="checkbox"/>
	Alison Stephenson CEO @ Gaptec	alisonstephenson@gaptec.biz			+1 (877) 451-3298		Gaptec	<input checked="" type="checkbox"/>
	Candace Leonard CEO @ Zaggie	candaceleonard@zaggie.info			+1 (934) 590-2323		Zaggie	<input checked="" type="checkbox"/>
	Kathie Bradshaw CEO @ Rooforia	kathiebradshaw@rooforia.us			+1 (990) 583-3613		Rooforia	<input type="checkbox"/>
	Cecelia Mcquire CEO @ Flum	cecelliamcquire@flum.name			+1 (805) 508-3016		Flum	<input checked="" type="checkbox"/>
	Teri Haley CEO @ Asimilne	terihaley@asimilne.me			+1 (953) 556-2884		Asimilne	<input checked="" type="checkbox"/>
	Janet Lowery CEO @ Plutorque	janetlowery@plutorque.tv			+1 (960) 508-2946		Plutorque	<input checked="" type="checkbox"/>
	Stark Dickson CEO @ Zypie	starkdickson@zypie.io			+1 (945) 535-3008		Zypie	<input checked="" type="checkbox"/>
	Santos Rose CEO @ Dreamia	santosrose@dreamia.com			+1 (993) 556-2770		Dreamia	<input type="checkbox"/>
	Samantha Clay CEO @ Confrenzy	samanthaclay@confrenzy.ca			+1 (848) 530-2851		Confrenzy	<input type="checkbox"/>
	Lawrence Thomas President @ Lazana	lawrence@lazana.org			+1 (889) 430-4211		Lazana	<input type="checkbox"/>
	Samantha Fields CEO @ Crystal Net	s.fields@crystalnet.com			+1 (990) 331-2453		Crystal Net	<input checked="" type="checkbox"/>



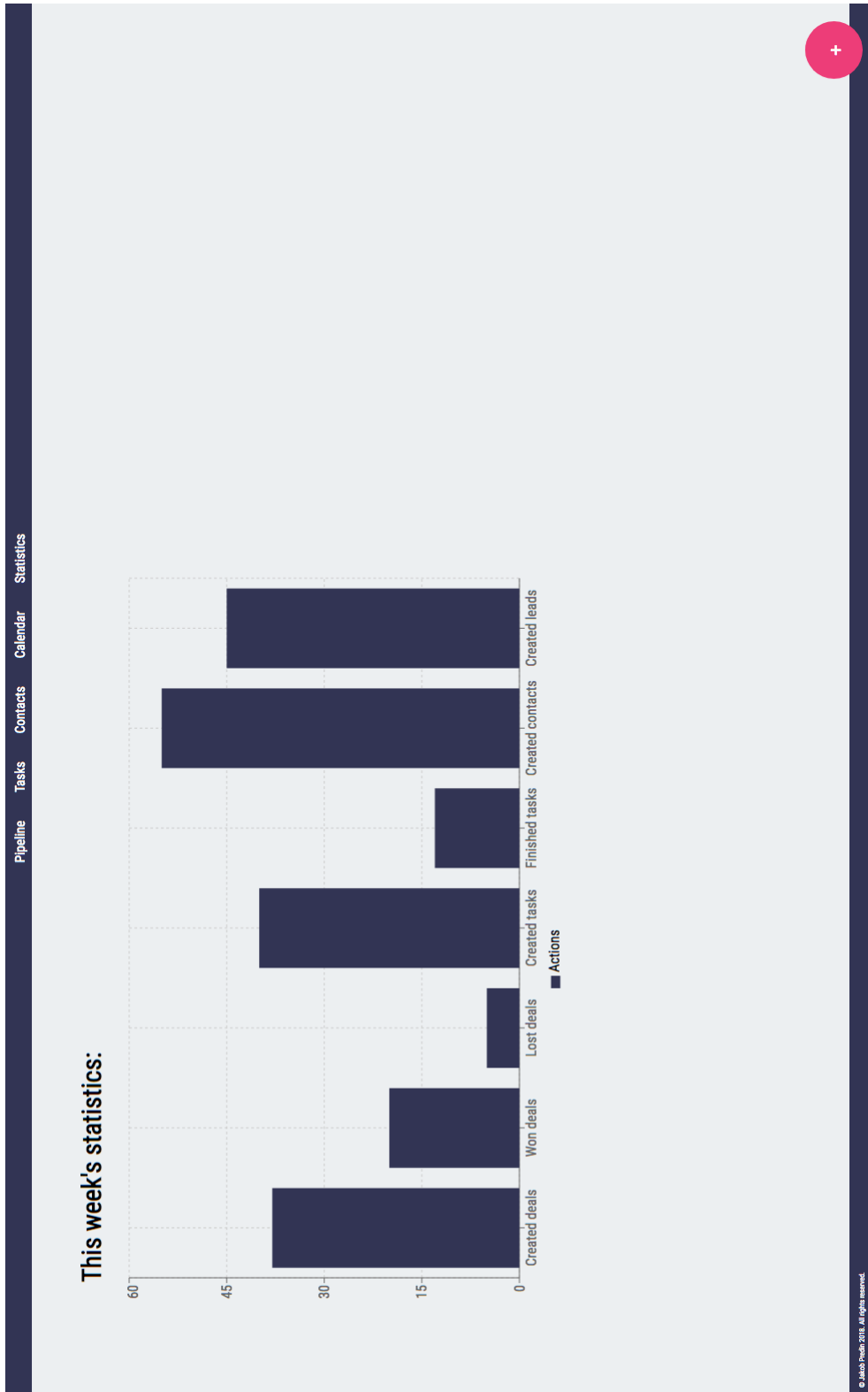
Silka A.3 Pogled upravljanja s stiki in potencialnimi strankami v sistemu.

The screenshot displays a CRM interface. At the top, a dark blue navigation bar contains the following menu items: Pipeline, Tasks, Contacts, Calendar, and Statistics. On the left, a contact card for George Solomon is shown, featuring a circular profile picture placeholder, the name 'George Solomon', and contact information: 'CEO @ Accidency', '+1 (920) 403-3291', and 'georgesolomon@accidency.name'. A red 'DELETE CONTACT' button is located at the bottom of the card. The main area of the screen shows a list of five tasks, each in a white box with a thin border. The tasks are: 'George Solomon was added to the task "Follow-up Lisa Austin", which has Very low importance.', 'George Solomon was added to the task "Call Violet Stevenson", which has Medium importance.', 'George Solomon was added to the task "Meet with Jeannine Campbell", which has High importance.', 'George Solomon was added to the task "Call Santos Gamble", which has Very low importance.', and 'George Solomon was added to the deal "Techmania deal" worth 1000\$ with a 92% win probability.'. A red circular button with a white plus sign is positioned in the top right corner of the interface. At the bottom right, a small copyright notice reads: '© 2016 Pivotal. All rights reserved.'

Slika A.4 Pogled posameznega stika v sistemu.



Slika A.5 Pogled pregleda aktivnosti in dogodkov z opravi v sistemu.



Slika A.6 Statistični pogled v sistemu.