

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Katarina Rovan

**Analiza modela PaaS in razvoj aplikacije na platformi
Salesforce**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Katarina Rovan

**Analiza modela PaaS in razvoj aplikacije v oblaku na
platformi Salesforce**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2017

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva – Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu lahko distribuira, predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.¹

¹ V dogovoru z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. *Creative Commons Napaka! Vira sklicevanja ni bilo mogoče najti.* in *GNU GPL Napaka! Vira sklicevanja ni bilo mogoče najti.* Zgornje besedilo je opis licence, ki ga po potrebi lahko tudi prilagodite. Če se kandidat odloči, da diplomskega dela ne bo izdal pod omenjenimi licencami, je potrebno zgornje besedilo spremeniti v naslednje: »Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.« **V obeh primerih pa iz končnega besedila odstranite to opombo.**

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge

Računalništvo v oblaku je postalo zelo popularen tehnološki trend in veliko poznavalcev področja verjame, da bo spremenil informacijsko tehnologijo. Prednosti računalništva v oblaku so visoka razpoložljivost, lahka dostopnost in prihranki pri stroških. V okviru diplomske naloge predstavite pojem PaaS in izdelajte mobilno ter spletno aplikacijo za vodenje zobozdravstvenega laboratorija. Predstavite potek izdelave take aplikacije z uporabo platforme Salesforce.

Zahvaljujem se družini in prijateljem za pomoč in podporo pri študiju. Zahvaljujem se mentorju viš. pred. dr. Aljažu Zrnecu za potrpežljivost in odzivnost. Zahvaljujem se tudi Luku Ivanšku za pomoč in podporo.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Računalništvo v oblaku	3
2.1	Javni oblak	4
2.2	Zasebni oblak	4
2.3	Hibridni oblak	4
2.4	Oblak skupnosti	5
Poglavje 3	Modeli storitev v oblaku	7
3.1	Infrastruktura kot storitev	8
3.2	Platforma kot storitev	8
3.3	Programska oprema kot storitev	8
3.4	Platforma kot storitev ponudnika Salesforce	9
3.5	Primerjava ponudnikov PaaS	11
3.5.1	Google App Engine	12
3.5.2	Amazon Web Services	13
3.5.3	Salesforce	14
3.6	Prednosti in slabosti razvoja aplikacije s PaaS	15
Poglavje 4	Platforma Salesforce	17
4.1	Aplikacijska logika	19
4.1.1	Izdelava aplikacij po principu izberi in klikni	19
4.1.2	Koda Apex	21
4.2	Varnost	22
4.2.1	Uporabniška varnost	22

4.2.2	Programska varnost.....	23
4.2.3	Varnostno ogrodje.....	23
4.3	Mobilna aplikacija.....	24
Poglavje 5	Razvoj aplikacije v oblaku na platformi Salesforce	25
5.1	Načrtovanje	25
5.2	Spoznavanje platforme Salesforce	27
5.3	Izdelava aplikacije.....	28
5.3.1	Objekti, ustvarjeni po meri.....	30
5.3.2	Uporabljeni standardni objekti.....	37
5.4	Mobilna aplikacija.....	41
5.5	Stroški razvoja aplikacije	41
Poglavje 6	Sklepne ugotovitve	43
6.1	Težave	43
6.2	Možne nadgradnje	43
Poglavje 7	Literatura.....	45

Kazalo slik

Slika 1: Vizualna predstavitev vrst računalništva v oblaku [13].....	3
Slika 2: Vizualni prikaz modelov storitev SaaS, PaaS in IaaS [49]	7
Slika 3: Logotip GAE [27]	13
Slika 4: Logotip AWS [26].....	14
Slika 5: Logotip Salesforce [33].....	14
Slika 6: Struktura skupne infrastrukture [36]	18
Slika 7: Urejevalnik objekta [45]	20
Slika 8: Urejanje v vrstici [44]	20
Slika 9: Diagram primerov uporabe	26
Slika 10: Shema podatkovnega modela izdelane aplikacije.....	29
Slika 11: Stran naročnikov	30
Slika 12: Stran produktov	32
Slika 13: Stran zaloge	33
Slika 14: Stran dobaviteljev.....	34
Slika 15: Obrazec za naročilo	35
Slika 16: Barvni ključ za izbiro barve zob [30].....	36
Slika 17: Stran z naročili	37
Slika 18: Ustvarjanje dogodkov za prikaz na koledarju	38
Slika 19: Primer poročila.....	39
Slika 20: Primer pregledne plošče	40

Kazalo tabel

Tabela 1: Primerjava PaaS ponudnikov.....	15
---	----

Seznam uporabljenih kratic

kratica	angleško	slovensko
IaaS	Infrastructure as a Service	infrastruktura kot storitev
PaaS	Platform as a Service	platforma kot storitev
SaaS	Software as a Service	programska oprema kot storitev
SAML	Security Assertion Markup Language	označevalni jezik za varnostne trditve
CRM	Customer Relationship Management	upravljanje odnosov s strankami
SOAP	Simple Object Access Protocol	protokol za spletne storitve
WSDL	Web Services Description Language	opisovalni jezik spletnih storitev
WS-I	Web Services-Interoperability	interoperabilne spletne storitve
API	Application Programming Interface	programski vmesnik
HTTPS	HyperText Transfer Protocol Secure	protokol, ki omogoča varno spletno povezavo
IDE	Integrated Development Enviroment	integrirano razvojno okolje
SDK	Software Development Kit	paket za razvoj programske opreme
RDS	Relational Database Service	relacijske podatkovne storitve
IT	Information Technology	informacijska tehnologija
SAP	Service Access Point	točka dostopa do storitve
REST	Representational State Transfer	predstavitveni prenos stanja
AWS	Amazon Web Services	spletne storitve podjetja Amazon

GAE

Google App Engine

aplikacijski strežnik, ki ga oskrbuje
podjetje Google

IP

Internet Protocol

internetni protokol

SOQL

Salesforce Object Query
Language

strukturiran povpraševalni jezik
platforme Salesforce za delo s
podatkovnimi bazami objektov

Povzetek

Naslov: Analiza modela PaaS in razvoj aplikacije v oblaku na platformi Salesforce

Glavni cilj diplomske naloge je predstaviti računalništvo v oblaku, analiza modela platforme kot storitve in izgradnja aplikacije na platformi Salesforce. V diplomski nalogi najprej predstavimo računalništvo v oblaku in opišemo glavne modele storitev v oblaku, sledi podrobnejši opis platforme kot storitve na platformi Salesforce. Opišemo dve v praksi pogosto uporabljeni platformi kot storitvi, ki ju primerjamo s platformo Salesforce. Pri razvoju aplikacije so opisane značilnosti platforme Salesforce, ki jih uporabljamo pri gradnji aplikacije. Opišemo še prednosti in slabosti gradnje aplikacije na platformi kot storitev, predstavimo aplikacijo za zobotehnični laboratorij in potek gradnje aplikacije. Na koncu zaključimo s končnimi ugotovitvami, kjer opišemo težave in možne nadgradnje aplikacije.

Ključne besede: Salesforce, računalništvo v oblaku, platforma kot storitev.

Abstract

Title: PaaS model analysis and application development on the Salesforce platform

The main goal of the thesis is to present cloud computing, analyse the platform model as a service and build application on the Salesforce platform. In the thesis we first present cloud computing and describe the main forms of cloud services. Follow the detailed descriptions of the platform as a service on the Salesforce platform. We also describe two other practices that are often used in practice and compare them with the Salesforce platform. Application development describes the features of Salesforce platform that we use when we develop our application. We also describe the advantages and the weakness of building applications on the platform. We present the application for dental lab and describe development. At the end we conclude with the final findings by describing the problems and possible application upgrades.

Keywords: Salesforce, cloud computing, platform as a service

Poglavje 1 Uvod

Računalništvo v oblaku se hitro razvija in nam nudi vedno več storitev po nizkih cenah. Storitve v oblaku, kot je Salesforce, omogočajo podjetjem, da se osredotočajo na svoje poslovanje, namesto na porabo sredstev za računalniško infrastrukturo in vzdrževanje. Računalništvo v oblaku podjetjem omogoča, da se izognejo stroškom IT (angl. Information Technology) ali jih vsaj zmanjšajo. Zagovorniki trdijo, da računalništvo v oblaku podjetjem omogoča hitrejše izvajanje in prenašanje aplikacij v produkcijsko okolje. Prav tako naj bi tudi zagotavljalo boljše upravljanje in manj vzdrževanja aplikacij [11].

V diplomski nalogi se osredotočimo na platformo Salesforce in potek razvoja spletne aplikacije na modelu platforme kot storitve, ki je namenjena razvijalcem. Salesforce se je začel razvijati kot "programska oprema kot storitev" za CRM (angl. Customer Relationship Management) in ponuja infrastrukturo, ki jo običajno potrebujemo za podjetja. Posebnost Salesforce platforme je tudi, da nam omogoča pisanje kode neposredno v oblaku.

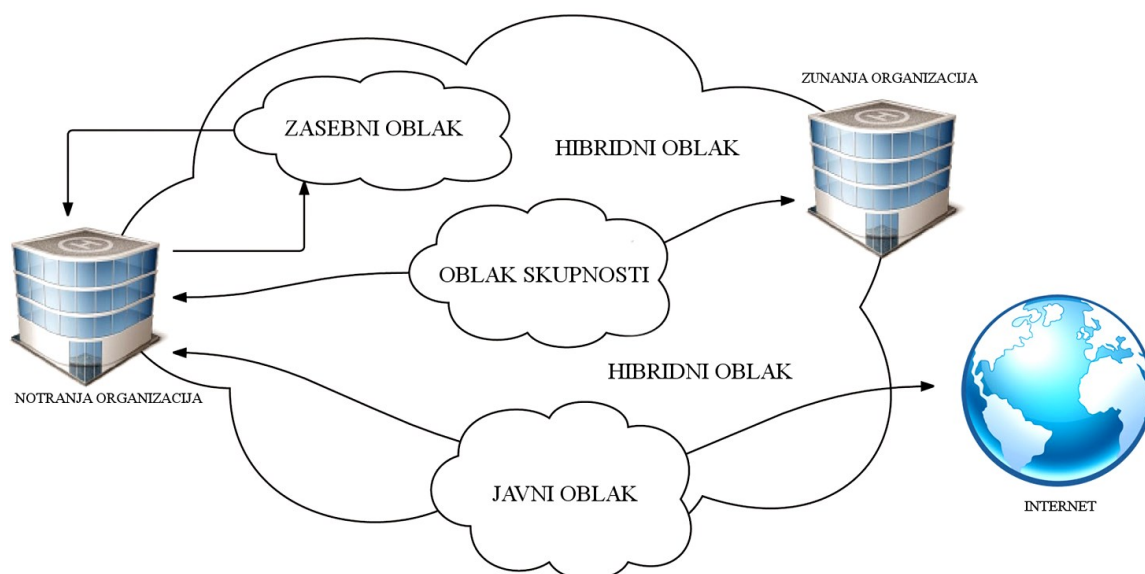
V prvem delu diplomske naloge opišemo, kaj je računalništvo v oblaku, in predstavimo modele storitev v oblaku. Opišemo model platforme kot storitve na platformi Salesforce in naredimo primerjavo različnih ponudnikov. Primerjava ponudnikov nam pomaga pri odločitvi, katerega ponudnika izbrati za našo aplikacijo.

V drugem delu diplomske naloge predstavimo platformo Salesforce. Opišemo glavne značilnosti Salesforce, varnostne vidike, značilnosti platforme in načine izdelave mobilne aplikacije. Potem se osredotočimo na razvoj aplikacije za zobotehnični laboratorij. Opišemo, kako smo se lotili izdelave in sam postopek izdelave.

Na koncu v ugotovitvah povzamemo vsebino diplomske naloge in opišemo, na katere težave smo naleteli tekom razvoja. Opišemo tudi možne nadgradnje aplikacije, ki smo jo izdelali.

Poglavje 2 Računalništvo v oblaku

Računalništvo v oblaku (angl. Cloud computing) je način zagotavljanja skupne rabe računalniških virov, ki vključujejo aplikacije, shranjevanje podatkov, mreženje in razvojne platforme. Poznamo različne vrste računalništva v oblaku glede na potrebe uporabnika. Glavne vrste oblakov so: javni, zasebni in hibridni; poznamo tudi oblak skupnosti. Slika 1 prikazuje vrste oblakov računalništva v oblaku.



Slika 1: Vizualna predstavitev vrst računalništva v oblaku [13]

Računalništvo v oblaku ima nekaj glavnih značilnosti, ki so [21]:

- **samopostrežba na zahtevo (angl. On-demand self-service);** stranka lahko avtomatsko izkorišča računalniške zmogljivosti, kot so procesiranje strežnika in čas obdelave brez interakcije ponudnika;
- **širok dostop do omrežja (angl. Broad network access);** zmogljivosti programske in strojne opreme v oblaku so na voljo prek omrežja in do njih lahko dostopamo preko različnih platform (mobilnih telefonov ter namiznih in tabličnih računalnikov);

- **zbiranje virov (angl. Resource pooling)**; programska in strojna oprema ponudnika sta združeni za več strank, uporabljajo večuporabniško (angl. Multi-tenant) arhitekturo;
- **hitra razširljivost (angl. Rapid elasticity)**; zmogljivosti lahko hitro pomanjšamo ali povečamo glede na zahteve, razširitev lahko kupimo kadarkoli;
- **izmerjena storitev (angl. Measured Service)**; sistem oblaka avtomatsko nadzira in optimizira uporabo virov. Uporaba virov se sproti nadzira in o njej se poroča, s čimer zagotovimo preglednost za stranko in ponudnika. Glavna prednost je, da plačujemo toliko, kot uporabljamo.

2.1 Javni oblak

Javni oblaki ponujajo uporabnikom storitve preko interneta. Namenjeni so različnim uporabnikom. Prednost javnega oblaka je v tem, da ni predhodnih stroškov, uporabnik za storitve plačuje sproti. Stroški storitev se lahko povečajo s prenosom podatkov v oblak in iz njega. Podjetja, ki uporabljajo javni oblak, so odvisna od zmožnosti in zanesljivosti izbranega ponudnika [7].

2.2 Zasebni oblak

Zasebni ali notranji oblak se nanaša na računalništvo v oblaku v zasebnih omrežjih. Zasebni oblak je namenjen točno določenemu uporabniku, običajno poslovnemu sistemu, in zagotavlja popoln nadzor nad podatki, varnostjo in kakovostjo storitve. Zasebne oblake je mogoče zgraditi in upravljati s strani lastnega IT ali ponudnika oblakov. Razširitev zasebnega oblaka lahko znatno poveča stroške [20].

2.3 Hibridni oblak

Hibridni oblak je kombinacija javnega in zasebnega oblaka. Podjetje ima zasebni oblak in v njem notranje vire, ki so pod nadzorom podjetja, zunanje vire pa priskrbi javni ponudnik storitev v oblaku. Hibridni oblak nudi podjetju shranjevanje občutljivih podatkov in specializirane storitve, ki morajo biti varne in so specifične za podjetje, hkrati pa imamo na voljo skalabilnost javnega oblaka [7].

2.4 Oblak skupnosti

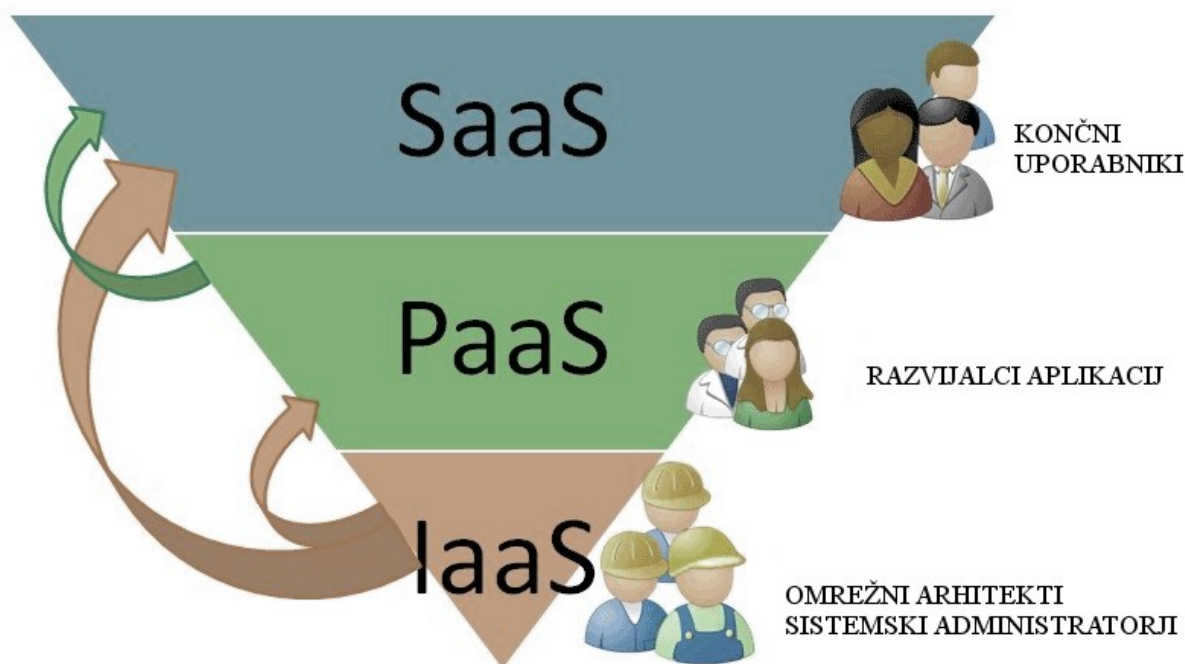
Oblak skupnosti deluje na omejenem številu uporabnikov s podobnimi potrebami, ki si delijo infrastrukturo oblaka. Cena storitev se razporedi med uporabnike, zato je oblak skupnosti bolj ekonomičen. Čeprav je javni oblak bolj ekonomičen od oblaka skupnosti, je v večini primerov oblak skupnosti z vidika varnosti, specializiranih storitev in zasebnosti boljši [7].

Poglavje 3 Modeli storitev v oblaku

Poznamo tri vrste modelov storitev v oblaku, in sicer:

- infrastruktura kot storitev – IaaS (ang. Infrastructure as a Service),
- platforma kot storitev – PaaS (ang. Platform as a Service) in
- programska oprema kot storitev – SaaS (ang. Software as a Service).

Modeli storitev v oblaku so namenjeni različnim uporabnikom. Najosnovnejši model storitve v oblaku je IaaS, ki je namenjen omrežnim arhitektom in sistemskim administratorjem, sledi PaaS, ki je namenjen razvijalcem aplikacij, in nazadnje SaaS, ki je namenjen končnim uporabnikom storitev. Grafični prikaz modelov storitev v oblaku se nahaja na sliki 2.



Slika 2: Vizualni prikaz modelov storitev SaaS, PaaS in IaaS [49]

3.1 Infrastruktura kot storitev

IaaS (angl. Infrastructure as a Service) se nanaša na računalniške vire kot storitve. Vključuje zagotovljeno procesorsko moč, prostor za shranjevanje podatkov in dostop do interneta. Uporabnik lahko na najet navidezni strežnik v oblaku namesti katerokoli programsko opremo in strankam ter osebju omogoči dostop. IaaS je najosnovnejša oblika modela storitev v oblaku. Uporabniki najemajo prostor za shranjevanje, požarni zid in drugo strojno in programsko opremo [12].

3.2 Platforma kot storitev

PaaS (angl. Platform as a Service) je nadgradnja IaaS, ki dodatno vsebuje operacijski sistem ter zahtevane storitve, kot so baze podatkov, datotečni sistemi in okolje za uporabo aplikacij. Na tem nivoju sta vključeni dve temeljni storitvi.

Storitev 1 predstavlja razvoj programske opreme ter njeno testiranje in delovanje v oblaku. Včasih je pisanje programske opreme preko omrežja v porazdeljenem računalniškem okolju za razvijalce predstavljalo veliko težavo. Izboljšanje pasovne širine omrežja je omogočilo dve rešitvi. Prva je orodje za razvoj na spletu, ki razvijalcem omogoča razvoj aplikacije na daljavo preko brskalnika in oddaljene konzole brez lokalne namestitve razvijalnih orodij. Druga je integracijska tehnologija lokalnih razvijalnih orodij in računalništva v oblaku, kar pomeni, da postavimo razvito aplikacijo neposredno v okolje računalništva v oblaku z lokalnimi razvijalnimi orodji.

Storitev 2 predstavlja obsežno porazdeljeno aplikacijsko okolje delovanja. Tako aplikacijsko okolje omogoča aplikaciji, da v celoti izkoristi računalniške in pomnilniške vire v oblaku [12].

3.3 Programska oprema kot storitev

SaaS (angl. Software as a Service) omogoča uporabnikom dostop do aplikacij v oblaku. Na nivoju SaaS ponudniki storitev ponujajo individualnim osebam in podjetjem uporabniške in industrijske aplikacije. Sistemi SaaS imajo naslednje značilnosti [4]:

- **dostopnost preko spletnega brskalnika;** SaaS ne zahteva namestitve programske opreme na lokalnem računalniku, vendar do nje dostopamo preko spletnega brskalnika;

- **razpoložljivost na zahtevo**; ko imamo enkrat dostop do programske opreme, ki temelji na programski opremi SaaS, moramo imeti možnost dostopa kjerkoli in kadarkoli;
- **plačilni pogoji na podlagi uporabe SaaS**; ne potrebujemo naložbe v infrastrukturo ali izvajanja zapletenih nastavitvev, torej nimamo velikih stroškov za namestitvev. Preprosto plačujemo za storitve, ki jih uporabljamo, ko jih ne potrebujemo več, pa s plačevanjem prenehamo;
- **minimalne zahteve IT**; konfiguracija in prilagoditev aplikacije, izdelane s sistemom SaaS, zahteva določen nivo tehničnega znanja. Če prilagoditev v glavnem zagotavlja ponudnik, ni potrebe po visokem nivoju tehničnega znanja razvijalca. Razvijalec se lahko osredotoči na razumevanje zahtev podjetja.

3.4 Platforma kot storitev ponudnika Salesforce

Najboljše storitve PaaS zagotavljajo popoln nabor orodij in storitev za preprost razvoj. Platforma Salesforce predstavlja sedem osnovnih storitev sistema PaaS [31]:

- **SDK** (angl. Software Development Kit); vedno več podjetij uporablja mobilne telefone, zato je pomembno, da nam platforma omogoča razvoj mobilnih aplikacij. Salesforce mobilni SDK je odprtokodni paket znanih tehnologij, kot so REST (angl. Representational State Transfer) API (angl. Application Programming Interface) in OAuth 2.0, kar nam poenostavi razvoj.
- **Usmerjenost v socialni in mobilni razvoj** (angl. Social and mobile build-in); socialno omrežje ni več omejeno samo za osebno uporabo, ampak se vse bolj integrira v poslovne kontekste, spodbuja produktivnost zaposlenih in vključuje stranke. Mnogi IT oddelki se trudijo zagotavljati najnovejše poslovne zahteve za socialni in mobilni razvoj. Uporaba okolja PaaS omogoča lažje izkoriščanje naših obstoječih veščin, kar omogoča lažje in hitrejšo posodobitve.
- **Obogateno okolje za razvijalce** (angl. Rich developer environment) in omogočanje razvijalcem razvoj uporabnih, privlačnih, uporabniško usmerjenih (angl. customer-facing) aplikacij, ki navdušijo in obdržijo uporabnike; kakovostne rešitve PaaS sistema nudijo razvijalcem vse, kar potrebujejo za hitro ponovitev, pomoč pri testiranju in razhroščevanje aplikacij. Obogateno okolje za razvijalce, ki ga ponuja platforma Salesforce, omogoča takojšnjo uveljavitev sprememb, lažje spreminjanje in

popoln nadzor, visoko zmogljivost in neodvisno prilagajanje za vsako komponento aplikacije.

- **Popolnoma upravljana infrastruktura v oblaku;** nudi nam popolno upravljano infrastrukturo, ki lahko presega naše trenutne potrebe, ker se v celoti upravlja. Infrastruktura v oblaku je varna in izpolnjuje zahteve podjetja o zasebnosti podatkov v skladu s predpisi podjetja.
- **Gradnja aplikacije po principu izberi in klikni** (angl. Point-and-click) je napredna rešitev PaaS sistema, ki omogoča uporabnikom razvoj lastnih aplikacij brez uporabe programske kode. Podpora za storitve na ravni podjetja vključuje postavitev strani objektov v aplikaciji po principu povleci in spusti (angl. Drag-and-drop), ustvarjanje polj (glej Poglavlje 5, Razvoj aplikacije v oblaku na platformi Salesforce) aplikacij v objektih z izberi in klikni in izdelavo poročevalskih tabel.
- **Večjezični razvoj;** večjezična podpora pomeni, da razvijalci uporabljajo in razvijajo aplikacije, napisane v jeziku, ki najbolj ustrezajo zahtevam podjetja. Platforme, kot je Salesforce, omogočajo pisanje kode v jezikih, kot so Python, Ruby on Rails, Node.js, Scala, Java, JVM in še več.
- **Tržnica aplikacij v oblaku** (angl. Cloud app marketplace) je spletna tržnica, ki omogoča dostop do aplikacij in storitev, ki jih razvijalci lahko uporabijo v razvoju za prilagoditev in razširitev storitve PaaS.

Storitve PaaS odpravljajo stroške in kompleksnost nakupa, konfiguriranja in upravljanja strojne in programske opreme, ki jih potrebujemo za razvoj aplikacije, kar nam daje številne tehnične in poslovne prednosti. Tehnične prednosti omogočajo razvojni ekipi, da se osredotoči na gradnjo aplikacij in storitev. Dober PaaS ponudnik nam lahko ponudi naslednje izboljšave na področju tehničnih rešitev [31]:

- **strokovno znanje** – ponudnik nam omogoči dostop do knjižnic, komponent aplikacije in druge pomoči;
- **hitrost** – razvoj aplikacij je hitrejši, saj razvijalci niso več odgovorni za vzdrževanje strojne in programske opreme, zato začnemo hitreje z uporabo sistema PaaS. Takoj po prijavi imamo dostop do orodij, knjižnic in predlog za hiter razvoj in izdajo aplikacije;

- **stroški** – začetni stroški so manjši, saj nam pred izdelavo aplikacije ni treba pripraviti okolja za razvoj aplikacij. Z izdelavo nove aplikacije nam ni treba vedno znova začeti od začetka, s tem si zmanjšamo stroške razvoja aplikacije;
- **obseg** – aplikacije za milijone povezanih naprav ustvarijo potencialne izzive razširljivosti in varnosti; čeprav je malo verjetno, da poslovna aplikacija doseže tolikšen obseg, imamo s sistemom PaaS možnost razširitve.

Uporaba integrirane platforme ponuja priložnost za izboljšanje poslovnih procesov, kot so potek dela, poročanje in socialni odzivi (angl. Feeds). Poslovne prednosti, pri katerih prihranimo čas, denar in napor, so [31]:

- **enostavna integracija s starimi sistemi** – lahko uporabljamo aplikacije, ki vsebujejo podatke iz naših obstoječih sistemov, kot so SAP (angl. Service Access point), Oracle in Microsoft;
- **podatki v realnem času** – ustvarimo lahko aplikacije, ki zagotavljajo podatke in posodobitve v realnem času. To zaposlenim in upravljalcem omogoča boljše poslovne odločitve;
- **enostavnejše vzdrževanje IT** – podjetje mora skrbeti samo za aplikacije, s tem zmanjša stroške IT;
- **skupni vpogledi (angl. Shared insights)** – veliki ponudniki, kot je Salesforce, se hitro odzovejo na potrebe uporabnikov in rešujejo pogosta vprašanja. S tem lahko hitro izkoristimo preizkušene in zaupanja vredne rešitve.

3.5 Primerjava ponudnikov PaaS

Imamo različna podjetja z različnimi zahtevami in proračunskimi omejitvami, kar se tiče razvoja aplikacij. Pri izbiri ponudnika PaaS moramo paziti na storitve in zmogljivosti, ki jih ponudnik nudi, da lahko zadovoljimo potrebe podjetja. Primerjali bomo platforme Salesforce, AWS (angl. Amazon Web Services) in GAE (angl. Google App Engine). Za AWS in GAE smo se odločili, ker predstavljata konkurenčni platformi ponudniku Salesforce [9]. Ponudnike za razvoj uporabnih in prilagodljivih aplikacij bomo primerjali po sledečih kriterijih:

- **Prilagodljivost in razširljivost;** pomembno je poiskati platformo, ki podpira več programskih jezikov in ogrodij, ki omogočajo prilagodljivost v našem kodirnem jeziku. Razširljiva infrastruktura v oblaku omogoča razvijalcem PaaS, da hitro

zagotovijo računalniško infrastrukturo in ponudijo visoko zmogljive aplikacije v oblaku končnim uporabnikom.

- **Shranjevanje podatkov;** razmisliti moramo o potrebah podjetja glede shranjevanja podatkov. Nekatere osnovne platforme omogočajo izbiro samo med relacijsko bazo ter bazo podatkov za shranjevanje na osnovi datotek (angl. file-based storage databases). Nekatere platforme ponujajo tudi druge možnosti, kot je uporaba NoSQL podatkovnih baz. Upoštevati moramo, da s količino shranjenih podatkov v oblaku naraščajo tudi stroški.
- **Integracija aplikacij;** ponudniki naj bi zagotavljali orodja za razvijalce in integracijo z aplikacijami, kot so programske platforme. Integracija s programi, kot je »Visual Studio« [48], lahko pomaga pospešiti postopek nalaganja in upravljanja s kodo.
- **Cena;** pri izbiri ponudnika je pomembna tudi cena ponujenih storitev.

3.5.1 Google App Engine

GAE (angl. Google App Engine) [22] predstavlja eno izmed kakovostnejših izbir in je z Googlovim imenom postala ena najbolj uporabljenih platform za podjetja vseh velikosti. Ko je podjetje izdelovalo platformo GAE, je vanjo vključilo nekaj API-jev, ki jih Google uporablja za ustvarjanje lastnih aplikacij. Prav tako ima nekaj infrastrukturnih značilnosti, ki so nam na dosegu v Googlovih infrastrukturnih virih preko sistema PaaS. Sistem PaaS vključuje Googlovo omrežje in omrežje za dostavo vsebin (angl. Content delivery network), ki predstavlja sistem strežnikov, razporejenih v omrežju. Vsebina izvirnega strežnika se podvoji in prenese na robne strežnike. Uporabniku zagotavlja uporabo najbližjega robnega strežnika.

GAE podpira več različnih programskih jezikov, kot so Java, Python, PHP in Googlov programski jezik GO, ne podpira pa toliko programskih jezikov kot konkurenčne platforme. Če za spletni razvoj uporabljamo Python ali Javo, so nam na voljo različni API-ji, dodatne storitve in vaje (angl. tutorials). Google App Engine ni najbolj prilagodljiv ponudnik storitev PaaS. Čeprav so nekatere storitve GAE namenjene profesionalnim uporabnikom, ki ustvarijo veliko prometa, ponuja platforma manj zanesljive rešitve kot konkurenca.

Model stroškov nam omogoča določiti željene funkcije, ki jih bomo plačevali. Podpora je ponujena preko knjižnic in drugih virov, vendar ne podpira podpore »v živo« brez doplačila dodatnega paketa podpore. Platforma je nameščena na lastni infrastrukturi.

Če nam je všeč kateri od štirih zgoraj omenjenih programskih jezikov, je GAE dobra izbira za razvoj aplikacij, pri katerem plačamo, kar uporabljamo, zato je primeren za aplikacije vseh obsegov. Logotip GAE je prikazan na sliki 3.



Slika 3: Logotip GAE [27]

3.5.2 Amazon Web Services

AWS (angl. Amazon Web Services) [3] se na prvi pogled zdi sistem IaaS, vendar gre za sistem PaaS, ki ponuja dodatne infrastrukturne funkcije. Čeprav so zmogljivosti IaaS na voljo v AWS, nam platforma omogoča razvijanje in postavitve aplikacij brez upravljanja in vzdrževanja strežnikov. Prednost je raznolikost kodirnih jezikov, ki jih lahko uporabljamo, je med najbolj prilagodljivimi sistemi na trgu.

Podatkovno bazo lahko ustvarimo na lastnih strežnikih ali jo postavimo na Amazonu z RDS (angl. Relational Database Service). Takšne infrastrukturne funkcije nam omogočajo, da svoje PaaS rešitve prilagajamo bolj kot v tradicionalnih PaaS modelih brez dodatnih infrastrukturnih funkcij.

Cene za naročanje in oblikovanje PaaS aplikacij se prilagajajo glede na naše potrebe. Imamo možnost preizkusa brezplačne različice storitve v oblaku. Z nakupom nam pripada usposabljanje, dodatna izobraževalna gradiva, dostop do forumov in stalna spletna in telefonska podpora.

AWS je prilagodljivi PaaS, ki je primeren za različne uporabnike in služi kot platforma PaaS z možnostjo večje kontrole IaaS. Logotip AWS je prikazan na sliki 4.



Slika 4: Logotip AWS [26]

3.5.3 Salesforce

Salesforce nam omogoča enostavno razvijanje mobilnih in namiznih aplikacij v oblaku. Platforma se je začela razvijati kot ponudnik storitev SaaS, potem je Salesforce svojo platformo prilagodil mobilnim razvijalcem in podjetjem. Ko je podjetje prešlo na model PaaS, je ohranilo nekaj prvotnih funkcij programske opreme za celovito platformo, kot so standardni objekti, ki so del platforme Salesforce in so opisani v nadaljevanju. Podjetje ponuja brezplačne predstavitve in preizkusno obdobje. Zgrajene aplikacije na platformi Salesforce so namenjene tako mobilni kot namizni uporabi. Omogočajo razvoj aplikacij po pristopu izberi in klikni, ki olajšuje in pospešuje razvoj aplikacij brez pisanja kode.

Mesečna naročnina se razlikuje glede na velikost aplikacije in namen uporabe aplikacije. Čeprav ni tako prilagodljiva kot druge platforme, je dodana komponenta SaaS z ustvarjalcem aplikacij, ki omogoča enostavno rešitev za nove uporabnike, saj ponuja standardne objekte, pripravljene za uporabo [9]. Logotip Salesforce je prikazan na sliki 5.



Slika 5: Logotip Salesforce [33]

	GAE	AWS	Salesforce
Podprti programski jeziki	Python, PHP, GO, Java	PHP, .NET, Node.js, Python, Ruby, Go	Phyton, Ruby on Rails, Node.js, Scala, Java, JVM
Podatkovni model	NoSQL	RDS	SOQL
Preizkusno obdobje	DA	DA	DA
Cena	Cena se izračuna na podlagi izbire zelenih storitev.	Cena se izračuna na podlagi izbire zelenih storitev.	Od 25 dolarjev na osebo na mesec za osnovno licenco.

Tabela 1: Primerjava PaaS ponudnikov

3.6 Prednosti in slabosti razvoja aplikacije s PaaS

Kot razvijalec spletnih aplikacij ali programske opreme je PaaS odlična rešitev za hiter razvoj prilagodljivih in razširljivih aplikacij. Prednosti razvoja na sistemu PaaS so [8]:

- **Hitro testiranje in postavitve;** razvojne skupine lahko preizkusijo različne konfiguracije z več napravami na različnih lokacijah. Izvajajo lahko teste izjemnih situacij in ocenijo uspešnost, združljivost in odziv na način, ki v lokalnem okolju ni mogoč. S hitrim testiranjem aplikacij je postavitve aplikacij hitrejša.
- **Dinamičen proces dodelitve virov (angl. Dynamic allocation);** na konkurenčnem trgu morajo IT ekipe imeti prožnost za hitro testiranje in postavitve novih lastnosti aplikacij, novih storitev na trgu, jih preizkusiti na majhnem delu strank pred izdajo. S PaaS in računalništvom v oblaku so takšne naloge postale lažje.
- **Povečana osredotočenost na podjetja in spodbujanje notranjega podjetništva;** podjetjem ni treba več vlagati v vzdrževanje in izbiro sistemov, lahko se osredotočijo na svojo osnovno dejavnost. S PaaS sta možna hitrejši razvoj in postavitve infrastrukture na spletu, kar spodbuja notranje podjetništvo. Notranje podjetništvo je način poslovanja podjetij in poudarja uvajanje novosti in spremembo rutin delovanja podjetja. Ko podjetja določijo proračun za aplikacijo v oblaku, lahko IT ekipa eksperimentira z računalniškimi viri v oblaku in določi, katere inovativne ideje bo razvila v aplikaciji.

Slabosti sistema PaaS [8]:

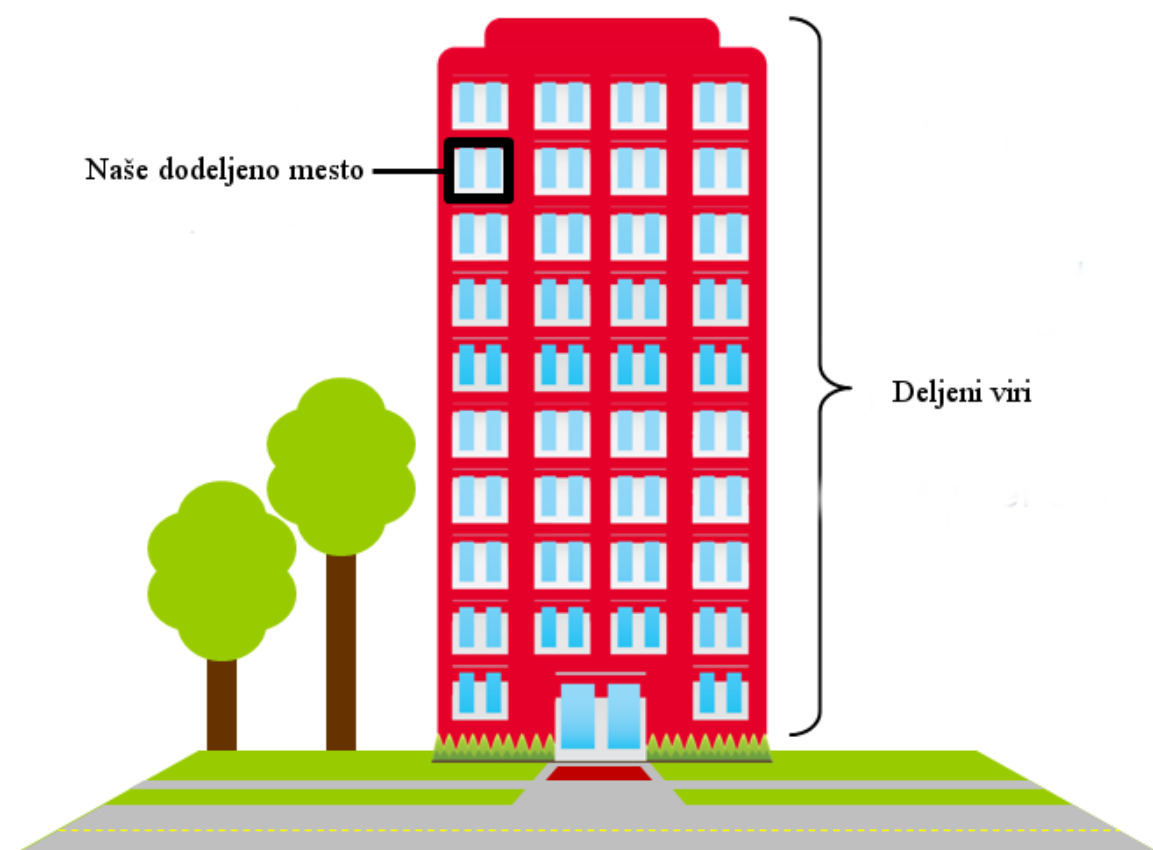
- **Varnost podatkov;** tako kot pri drugih rešitvah računalništva v oblaku mnoga podjetja še vedno ne zaupajo ravni varnosti podatkov. Nekaterim podjetjem je treba zagotoviti skladnost z vsemi veljavnimi predpisi o varnosti, zasebnosti in hrambi podatkov, preden se odločijo za sistem PaaS.
- **Omejena prilagodljivost;** PaaS rešitve ne morejo ustrezati prožnosti, ki jih ima sistem IaaS, kjer lahko ustvarimo in brišemo več virtualnih naprav. V primerjavi s SaaS PaaS ne predstavlja celotnega izdelka na način, kot ga SaaS, torej podjetje še vedno potrebuje razvojne napore za načrtovanje, ustvarjanje in testiranje aplikacij.
- **Problem integracije z lastnimi sistemi in aplikacijami;** integracija storitev PaaS z ostalimi lastnimi sistemi in aplikacijami nam lahko povzroči večjo kompleksnost pri integraciji.

Poglavje 4 Platforma Salesforce

Salesforce je zgrajen na Force.com platformi, ki nam omogoča gradnjo, deljenje in poganjanje aplikacij neposredno v oblaku. Ker nam ni treba kupiti strežnikov in programske opreme za razvoj, se lahko osredotočimo le na gradnjo aplikacij. Platforma Salesforce samodejno prilagaja, nastavi in izdela rezervne kopije podatkov. Večina ponudnikov računalništva v oblaku omogoča prenos programske kode iz lokalnega računalnika v oblak, Force.com pa nam omogoča, da programsko kodo pišemo neposredno v oblaku. Force.com je drugačen od drugih PaaS storitev, osredotoča se na poslovne aplikacije in je del Salesforce.com, ki pa se je začel kot SaaS za CRM. Sistem CRM je univerzalna rešitev problemov v odnosih s strankami in bo brez dvoma ostal glavni faktor za poslovanje v podjetjih. Platforma Salesforce ponuja standardni objekt chatter kot del sistema CRM. Chatter [10] je programska rešitev za zaposlene, kjer se zaposleni povežejo, sodelujejo, izmenjujejo datoteke in strokovno znanje. Vse omenjene funkcionalnosti potekajo v realnem času [35,29]. Značilnosti Force.com sistema v oblaku so :

- večuporabniška arhitektura,
- avtomatske posodobitve,
- naročniški model,
- razširljivost,
- posodobitve in vzdrževanje.

Večuporabniška arhitektura omogoča uporabnikom, da si delijo skupne vire. Podatki uporabnikov so izolirani in ostanejo nevidni drugim uporabnikom. Vsak uporabnik dobi svojo celo verzijo programske konfiguracije in podatkov. Programska oprema je virtualno razdeljena na različne instance. Tako kot si stanovalci bloka delijo uporabo elektrike in vode, si vsak najemnik deli skupne vire in podatkovno bazo. Struktura skupne arhitekture je prikazana na sliki 6.



Slika 6: Struktura skupne infrastrukture [36]

Avtomatične posodobitve predstavljajo posodobitve platforme Salesforce, ki so na voljo avtomatično za vse uporabnike [34].

Naročniški model je model, ki opredeljuje uporabniško licenco. Ta določa, do katerih storitev platforme Salesforce lahko uporabnik dostopa. Z uporabniško licenco uporabnik zgradi želeno aplikacijo. Uporabnik lahko dokupuje dostop do storitev platforme Salesforce in s tem spreminja uporabniško licenco. Storitve platforme Salesforce so omejene glede na licenco. Osnovna licenca nam ne ponuja sledečih storitev: omejitve prostora za shranjevanje podatkov, števila objektov, ustvarjenih po meri, omejitve prostora za shranjevanje datotek itd. Model omogoča nizek začetni vložek in možnost dokupovanja storitev platforme Salesforce s fiksnimi cenami in s tem tudi predvidenimi stroški za nadaljnji razvoj.

Osnova platforme Force.com oblikuje večuporabniško jedro, kjer si vsi uporabniki delijo skupno arhitekturo. Večuporabniško jedro je že testirano in se izvaja za več uporabnikov hkrati. Zaradi večuporabniškega jedra je v primeru rasti podjetja vedno prostor za **razširitev** aplikacij za več uporabnikov brez dodatnih skrbi za uravnoteženje bremena in omejitve

podatkov. Force.com omogoča shranjevalni prostor glede na potrebe uporabnika, zato se s povečevanjem števila uporabnikov povečuje tudi prostor za shranjevanje podatkov.

Force.com izda tri **posodobitve** vsako leto. Nove izdaje vsebujejo nove storitve platforme Salesforce in Force.com z izbranimi najboljšimi idejami iz IdeaExchange [23]. IdeaExchange je skupnost uporabnikov Salesforce, kjer uporabniki predlagajo ideje za nove storitve in za njih glasujejo. Najboljše ideje za nove storitve se izbere in izda v naslednjih posodobitvah [35].

4.1 Aplikacijska logika

Platforma Force.com omogoča izdelavo aplikacij na dva načina:

- po principu izberi in klikni (angl. Point-and-click) in
- s programskim jezikom Apex.

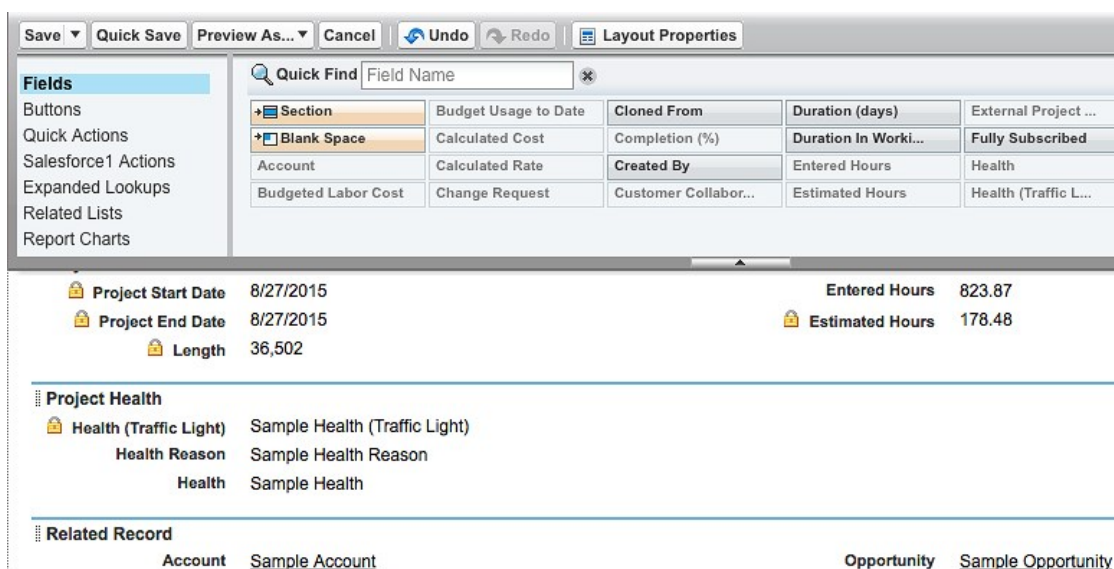
4.1.1 Izdelava aplikacij po principu izberi in klikni

Izdelava aplikacij po principu izberi in klikni nam omogoča razvoj novih funkcij ali prilagajanje aplikacij brez programske kode. S pristopom izdelave aplikacij po principu izberi in klikni smo omejeni pri načrtovanju aplikacij. Omejitve pri načrtovanju aplikacij rešimo z uporabo programskega jezika Apex. Prednosti pristopa gradnje aplikacij po principu izberi in klikni so:

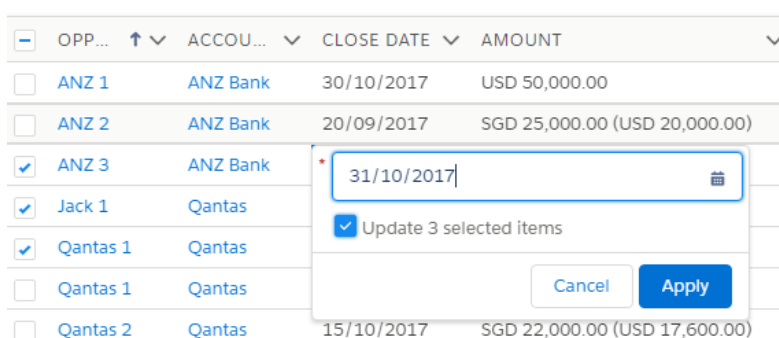
- **stroški in čas** – gradnja aplikacij po principu izberi in klikni je hitra, saj nam ni treba pisati programske kode, testnih primerov, nimamo skrbi s prenosom programske kode v repozitorij, kontrol verzij programske kode itd.;
- **vzdrževanje** – vzdrževanje je ena večjih razlik, saj nimamo programske kode, ki jo je treba nadgrajevati ter potrebe po vzdrževanju avtomatskih testov;
- **posodobitev poti (angl. Update path)** – z izdelavo aplikacij po principu izberi in klikni se nam aplikacija z izdajo nove različice Salesforce avtomatsko posodobi. Aplikacije, zgrajene s programsko kodo, nam še vedno delujejo, vendar moramo kodo po izdaji nove različice Salesforce prilagoditi za nove storitve;
- **uporabnost in uporabniška produktivnost** – gradnja aplikacij po principu izberi in klikni omogoči bolj konsistentno uporabnost aplikacije, to pomeni, da enak izgled in funkcije uporabljamo na vseh mestih v aplikaciji. Izgubimo možnost spreminjanja

postavitve strani (angl. Page layout) objekta z urejevalnikom in v pogledih izgubimo možnost za urejanje v vrstici. Urejevalnik objekta je prikazan na sliki 7, primer urejanja v vrstici je prikazan na sliki 8;

- **kompleksnost in razširljivost** – z gradnjo aplikacij po principu izberi in klikni lahko zgradimo kompleksno aplikacijo brez uporabe programske kode. Z gradnjo aplikacij brez programske kode, po principu izberi in klikni, imamo omejitve pri načrtovanju, vendar nimamo Governor omejitev (angl. Governor limits). Governor omejitve so obrazložene v nadaljevanju.



Slika 7: Urejevalnik objekta [45]



Slika 8: Urejanje v vrstici [44]

Salesforce ponuja bogat nabor storitev za razvoj aplikacij. Pred začetkom razvoja se je treba seznaniti z lastnimi zmožnostmi platforme Salesforce. Njihova uporaba nam prihrani čas in stroške vzdrževanja, včasih tudi poveča uporabnost aplikacij [47].

Governor omejitve [6] so omejitve na platformi Salesforce, ki nas silijo v pisanje učinkovite in razširljive programske kode. Ker programski jezik Apex deluje v večuporabniškem okolju, omejitve zagotavljajo, da uporabnikova Apex programska koda ali procesi ne zasedejo vseh skupnih virov. Poznamo omejitve, ki se navezujejo na:

- Apex transakcije (angl. Per-transaction Apex),
- certificirane upravljane pakete (angl. Certified managed packages), to so komponente aplikacij, ki so dostopne na tržnici aplikacij,
- Apex omejitve glede na Force.com platformo,
- statične Apex omejitve,
- različne Apex meje (angl. Miscellaneous Apex Limits), kot so omejitve za SOQL (angl. Salesforce Object Query Language) in
- omejitev števila obvestil (angl. Push Notification Limits).

4.1.2 Koda Apex

Apex je lastniški programski jezik, ki ga lahko izvajamo samo na platformi Salesforce. Ta ni odprtokodna in se izvaja samo na Salesforce strežnikih [25].

Platforma Salesforce ima konzolo za razvijalce [17], ki je integrirano razvojno okolje z zbirko orodij in jo uporabljamo za ustvarjanje, odpravljanje napak in preizkušanje aplikacij. Konzola nam pomaga pri številnih razvojnih nalogah:

- **razhroščevanje in odpravljanje napak** – konzola ponuja priročen nabor orodij za učinkovito razhroščevanje in odpravljanje napak v aplikacijah;
- **urejanje in navigacija izvorne kode** – konzola omogoča brskanje, odpiranje, urejanje in ustvarjanje datotek izvorne programske kode;
- **testiranje in preverjanje zmogljivosti** – konzola ima številne funkcije, ki so namenjene testiranju programske kode in analizi zmogljivosti, ker preko dnevnika spremljamo delovanje aplikacije.

Force.com IDE (angl. Integrated Development Enviroment) je integrirano razvojno okolje za ustvarjanje, spreminjanje in postavitve Force.com aplikacij v produkcijsko okolje [19].

4.2 Varnost

Force.com temelji na robustni in fleksibilni varnostni arhitekturi, ki zagotavlja visoko stopnjo nadzora nad uporabniki, omrežjem in podatki.

4.2.1 Uporabniška varnost

Večina uporabnikov se overi na strani za prijavo [32]. Znotraj podjetja pa se uporablja eden od dveh načinov overitve uporabnika.

1. **Delegirana overitev**, kjer se uporabnik prijavi v platformo na strani za prijavo, vendar platforma preko spletne storitve (angl. Web Service) posreduje ime in geslo zunanjemu overitvenemu pooblaščenцу. Pooblaščenec odobritev prenese nazaj na platformo in uporabniku omogoči prijavo. Za uporabo delegirane overitve moramo stopiti v stik s Salesforce, ki to funkcijo omogoči.
2. **Overitev z uporabo protokola SAML (angl. Security Assertion Markup Language)**, kjer zahteva za overitev uporabnika poteka s pomočjo protokola SAML. Ponudnik identitete je v tem primeru naše podjetje, ki overja uporabnike in vrne žeton. Žeton se prenese na platformo in uporabnik se preveri na podlagi ustreznega zapisa ponudnika identitete. SAML lahko konfiguriramo v varnostnih nastavitvah platforme Salesforce.

Uporabniška konfiguracija določa, kdo se lahko prijavi v sistem, določa tudi varnostne omejitve na omrežju, ki omejujejo čas in lokacijo prijave uporabnika. Platforma Salesforce omogoča dve možnosti za omejitev dostopa glede na lokacijo.

1. Uporabniku omogoča dostop iz zaupanja vrednih lokacij, ki jih uporabnik nastavi kot niz IP (angl. Internet Protocol) naslovov. Če se uporabnik želi prijaviti z računalnikom zunaj zaupanja vrednih lokacij, mu sistem pošlje elektronsko pošto s povezavo, preko katere doda novo zaupanja vredno lokacijo.
2. V uporabniškem profilu obstaja možnost nastavitve strožjih omejitev, s katerimi nastavimo zaupanja vredne lokacije kot niz IP naslovov. Uporabniku je omogočen dostop samo iz zaupanja vrednih lokacij.

Uporabnike lahko omejimo tudi z določitvijo časa, kdaj lahko dostopajo do aplikacije. Omejitve na podlagi IP naslovov se uporabljajo za zaščito pred napadom z ribarjenjem (angl. Phishing attack).

Po uspešni prijavi se ustvari seja, ki jo lahko nadzorujemo in v nastavitvah nastavljamo čas prekinitve seje. V nastavitvah lahko zahtevamo, da so vse strani vedno dostopne z varno povezavo preko protokola HTTPS (angl. HyperText Transfer Protocol Secure). Uporaba varnih povezav preko protokola HTTPS je v platformi Salesforce privzeto določena.

Zadnji vidik uporabniške varnosti je revizija. V nastavitvah lahko spremljamo zadnje prijave v aplikacijo ali prenesemo informacije o prijavah za zadnjih šest mesecev, shranjene so informacije o IP naslovih, brskalnikih in podobno. Revizijo je mogoče vključiti tudi za objekte ali posamezna polja [5].

4.2.2 Programska varnost

Veliko orodij, zgrajenih okoli platforme Salesforce, uporabljajo SOAP (angl. Simple Object Access Protocol) API, zato zahtevajo uporabo varnostnih žetonov. Uporaba varnostnih žetonov je odvisna od tega, ali je IP-naslov, s katerim se povezujemo v aplikacijo, zaupanja vreden. SOAP API se izvaja tako, da ustreza naslednjim specifikacijam: SOAP 1.1, WSDL (angl. Web Service Description Language) 1.1 in WS-I (angl. Web Services-Interoperability) Basic Profile 1.1 [37]. SOAP API podpira tudi preverjanje z uporabo OAuth 2.0 [28], ki je odprt protokol. Spletni strani mogoča dostop do virov druge spletne strani brez izpostavitve uporabniške poverilnice (angl. credential). OAuth omogoča dostop do virov z varnostnimi žetoni za določen čas [5].

4.2.3 Varnostno ogrodje

Platforma za varnost uporablja varnostno ogrodje (angl. Platform security Framework), s katerim določamo pravice dostopa do aplikacije za uporabnike in dostop do komponent platforme Salesforce, kot so objekti in zavihki. Področja varnosti na platformi Salesforce so:

- sistemske pravice (angl. System permission),
- pravice za komponente (angl. Component permissions) in
- souporaba na podlagi zapisa podatkov v objektih (angl. Record-based sharing).

Sistemske pravice določajo opravljanje nalog, kot so spremembe vseh podatkov, in veljajo za vse aplikacije. Sistemske pravice dodeljujemo uporabnikom. Pod sistemske pravice določamo administrativne pravice. Običajno jih določi skrbnik, ki izbere dostop sistema, prav tako določimo tudi pravice za poročila in pravice za manipulacijo s podatki.

Platforma Salesforce nam omogoča nastavitve pravic za posamezne komponente. Za vse objekte lahko nastavimo pravice za branje, ustvarjanje, urejanje in brisanje. Pravice lahko določamo tudi posameznim poljem v aplikaciji.

Souporaba na podlagi zapisa je zmožnost dodeljevanja pravic posameznim zapisom znotraj objekta. Dostop do posameznih zapisov lahko onemogočimo, omogočimo dostop samo za pisanje, dostop samo za branje itd. Platforma ponuja privzete vrednosti za celotno podjetje [5].

4.3 Mobilna aplikacija

Salesforce ponuja dva načina za izdelavo mobilne aplikacije, in sicer [1]:

- **aplikacija Salesforce**; z izgradnjo aplikacije s Salesforce platformo se avtomatsko ustvari mobilna različica aplikacije, ki je na voljo v trgovini AppStore in Google Play. Mobilna aplikacija ponuja najhitrejši način izdelave in uporabe aplikacije skrbnikom in razvijalcem;
- **Salesforce Mobile SDK** da razvijalcem možnost izdelave samostojnih mobilnih aplikacij po meri, ki jih distribuiramo preko trgovine AppStore ali Google Play.

Poglavje 5 Razvoj aplikacije v oblaku na platformi Salesforce

Pred začetkom razvoja aplikacije smo se morali odločiti, kakšne vrste aplikacijo bomo razvili. Platforma Salesforce je usmerjena v razvoj aplikacij za podjetja, zato smo se odločili za razvoj aplikacije za zobotehnični laboratorij. Večina zobotehničnih laboratorijev vodi svojo evidenco preko programa Microsoft Excel ali s kartotekami, ki jih shranjujejo v papirni obliki. S takšnim načinom dela je težko voditi evidenco, saj nimamo dobrega pregleda nad delom in statistiko.

V platformo Salesforce smo se prijavili kot razvijalci. Izdaja za razvijalce (angl. Developer Edition) je brezplačna kopija okolja podjetniške izdaje, s katero nam Salesforce omogoča brezplačni takojšnji razvoj in testiranje izdelane aplikacije [18]. Salesforce nam omogoča razvoj aplikacij v Salesforce Classic okolju ali okolju Lightning Experience [16]. Odločili smo se uporabiti Lightning Experience, ker je sodobnejši in vizualno privlačnejši. Ni vsaka Salesforce Classic funkcija podprta v Lightning Experience, vendar lahko enostavno preklapljammo med obema okoljema. Aplikacijo smo poimenovali MySmile in sledilo je načrtovanje aplikacije. Aplikacija je dostopna preko prijave z uporabniškim imenom in geslom na spletni strani Salesforce.

5.1 Načrtovanje

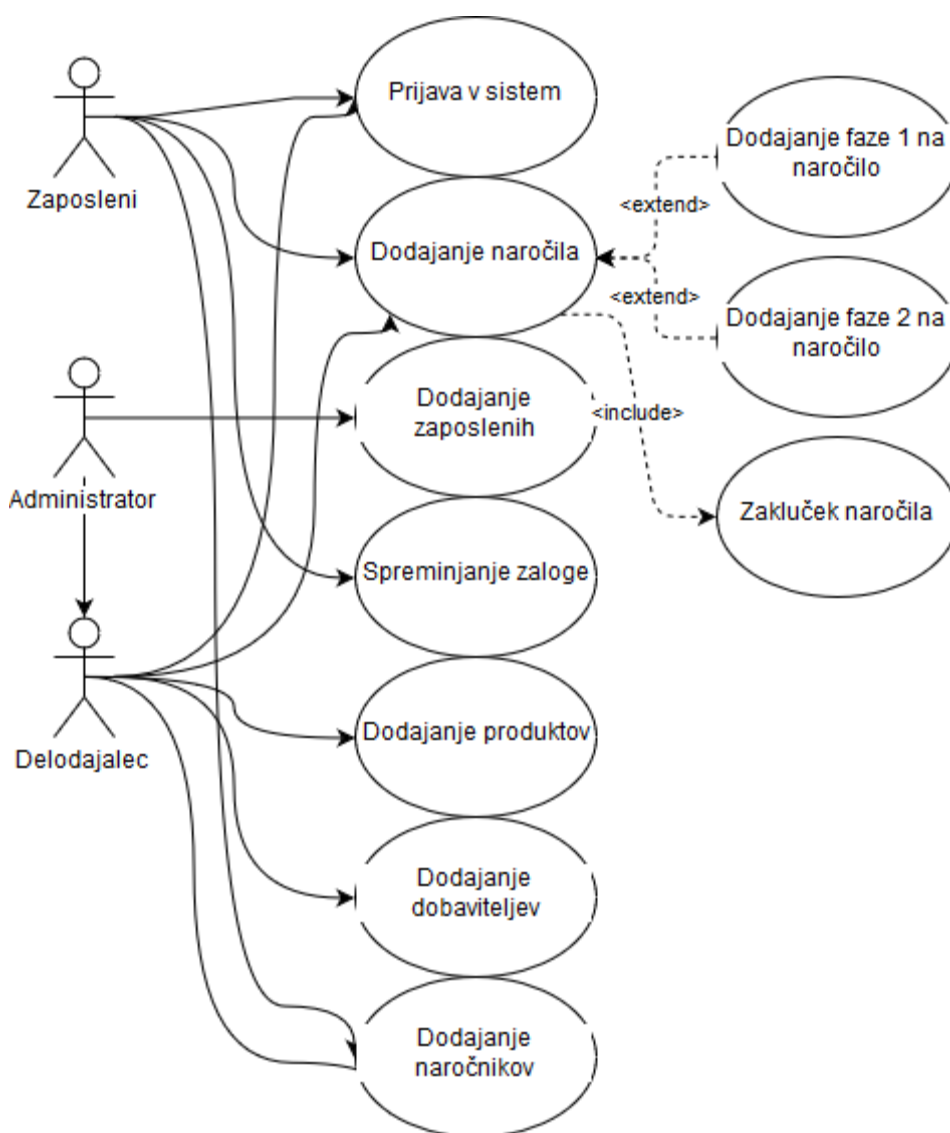
V fazi načrtovanja smo se pozanimali, kako poteka delo v zobotehničnem laboratoriju. V Sloveniji imamo veliko manjših zobotehničnih laboratorijev, zato v izdelani aplikaciji ni veliko definiranih vlog. Vloga določa pravice uporabnikom v aplikaciji. Izdelani aplikaciji smo določili sledeče vloge: zaposleni, vodja zobotehničnega laboratorija in administrator. Raziskali smo potek izdelave izdelka od naročila do zaključka. Potek izdelave je sledeč:

1. faza 1: zobozdravstvena ordinacija odda naročilo, ki zajema odtis delovnega področja in nasprotnega zobnega loka, registrat ugriza in izbrano barvo zob;
2. faza 2: zobni tehnik pripravi delovne modele, na katerih je izdelal kovinsko ogrodje pri fiksni protetiki ali po določenih predpisih postavil zobe v snemni protetiki.

Kovinsko ogrodje ali samo postavitev je treba preizkusiti v ustih pacienta in preveriti tesnjenje kovinskega ogrodja in linije ter ravnine pri vsakem pacientu. V tej fazi preverimo ustreznost izdelka določenim kriterijem in se odločimo, ali nadaljujemo z izdelavo izdelka ali ponovimo prvo fazo izdelave;

3. faza 3: vstavitev fiksoprotetičnega ali snemnoprotetičnega izdelka v usta pacienta, prilagoditev na nasprotni zobni lok in kontrola barve izdelka. Običajno v tej fazi pacient zapusti zobno ordinacijo s končnim izdelkom in obišče ordinacijo samo na rednih letnih pregledih.

Diagram primerov uporabe, ki prikazuje funkcionalnosti aplikacije, je prikazan na sliki 7.



Slika 9: Diagram primerov uporabe

5.2 Spoznavanje platforme Salesforce

Pri izdelavi aplikacije smo si pomagali s Trailhead [40]. Trailhead omogoča razvijalcem in administratorjem vodeno učno pot skozi ključne značilnosti platforme Salesforce. Z uporabo interaktivnih spletnih vadnic nam pomaga pri učenju. Spletnemu mestu stalno dodajajo nove funkcionalnosti in nove vsebine. Nekatere ključne funkcionalnosti Trailheada so [39]:

- **uporabniku prilagodljiv tempo (angl. Self-paced) učenja na spletu;** Trailhead uporablja model interaktivnega učenja, podobno kot spletna mesta Code school [14], Codecademy [15] in Udacity [41];
- **učne poti;** vsebina Trailhead je urejena v hierarhiji s sledmi, moduli in enotami ter predstavljena v določenem zaporedju. Stranke imajo vnaprej določeno pot, ki ji sledijo glede na izbrano vajo;
- **modularna vsebina, ki se povezuje glede na izbrano vajo;** vsaka vaja je sestavljena iz kratkih enot, ki trajajo od deset do petnajst minut. Vsebina je preprosta, samozadostna in neposredno uporabna za čim boljše učno izkušnjo;
- **interaktivne ocene;** a koncu vsake enote preko izziva preverimo, kaj smo se naučili, izziv je lahko določena naloga ali vprašanje;
- **igrifikacija (angl. Gamification);** za uspešno dokončano enoto ali modul prejmemo točke in značke, ki so prikazane v našem uporabniškem profilu.

5.3 Izdelava aplikacije

Na platformi Salesforce so objekti uporabljeni za shranjevanje podatkov. Objekti delujejo podobno kot tabele v podatkovni bazi. Objekti so lahko med seboj povezani, s povezavo omogočimo uporabo podatkov iz povezanega objekta. Objekti na platformi Salesforce vsebujejo polja (angl. fields) in zapise (angl. records). Polja v objektu se uporabljajo kot stolpci v podatkovni bazi, zapisi v objektu pa kot vrstice v podatkovni bazi. Poznamo sledeče objekte:

- **standardni objekti (angl. Standard objects)** – pogosto uporabljeni objekti, ki so vključeni v platformi Salesforce;
- **objekti po meri (angl. Custom objects)** – ustvarimo jih po potrebi za shranjevanje bolj specifičnih informacij aplikacij, za katere ne moremo uporabiti standardnih objektov.

Shema podatkovnega modela prikazuje objekte v aplikaciji, ki jim lahko dodajamo in spreminjamo polja ter zapise [42]. Shema podatkovnega modela izdelane aplikacije je prikazana na sliki 8.



Slika 10: Shema podatkovnega modela izdelane aplikacije

Objekti v izdelani aplikaciji, ki so ustvarjeni po meri, so:

- naročnik (angl. Client),
- produkt (angl. Product),
- zaloga (angl. Supply),
- dobavitelj (angl. Supplier) in
- naročilo (angl. Order) .

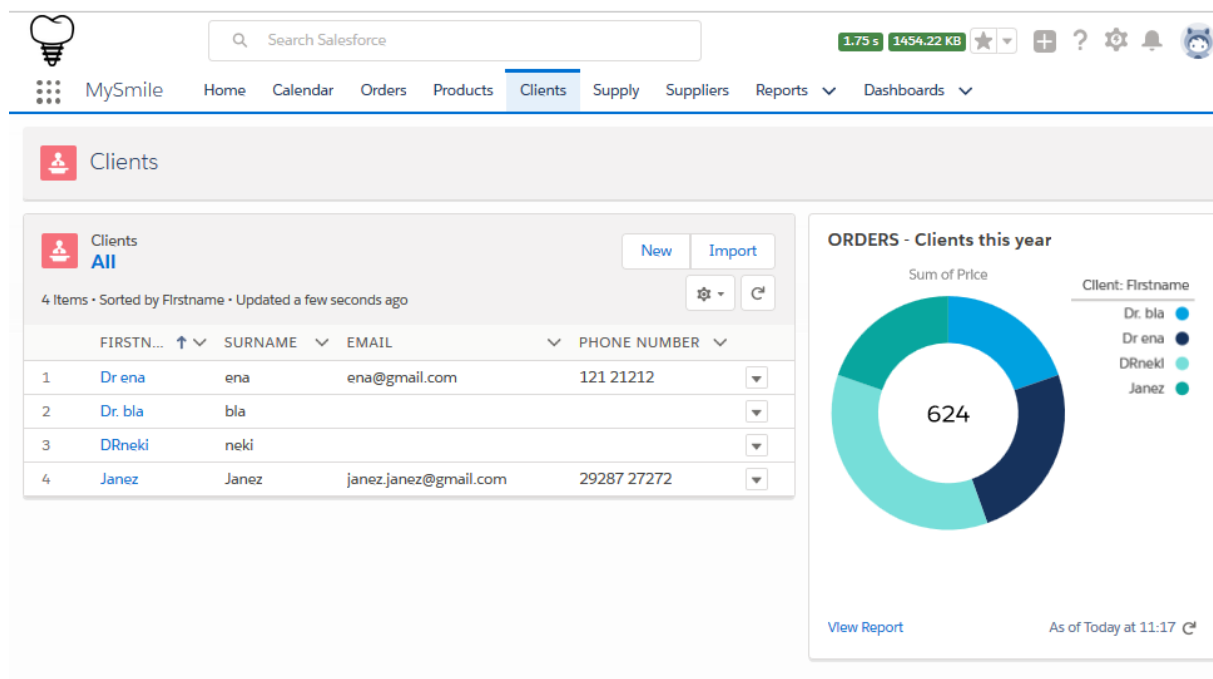
Uporabljeni standardni objekti v izdelani aplikaciji so:

- koledar (angl. Calendar),
- poročila (angl. Reports) in
- pregledne plošče (angl. Dashboards).

5.3.1 Objekti, ustvarjeni po meri

- **Naročnik**

V objektu naročnik so zapisi naročnikov, v našem primeru so to zobozdravniki, saj v zobotehničnem laboratoriju naročilo odda zobozdravnik. V objekt vnesemo vse zobozdravnike, s katerimi zobotehnični laboratorij sodeluje. To je pomemben podatek, prikazan na naročilu. Zobozdravnike lahko dodajajo tako zaposleni kot delodajalec. Na strani zobozdravnikov je prikazana statistika zobozdravnikov, kjer nam grafikon prikazuje odstotek opravljenih naročil posameznega zobozdravnika ter končno ceno vseh naročil zobozdravnika. Grafikone v aplikaciji prikazujemo s pomočjo standardnega objekta poročil, ki je opisan v nadaljevanju in nam na podlagi izdelanih poročil omogoča zgraditi grafikon. Stran z naročniki je prikazana na sliki 8.



The screenshot displays the Salesforce interface for the 'Clients' object. At the top, there is a search bar and navigation tabs including 'MySmile', 'Home', 'Calendar', 'Orders', 'Products', 'Clients', 'Supply', 'Suppliers', 'Reports', and 'Dashboards'. The 'Clients' tab is active, showing a list of 4 items sorted by Firstname. The table below lists the clients with their first names, surnames, emails, and phone numbers.

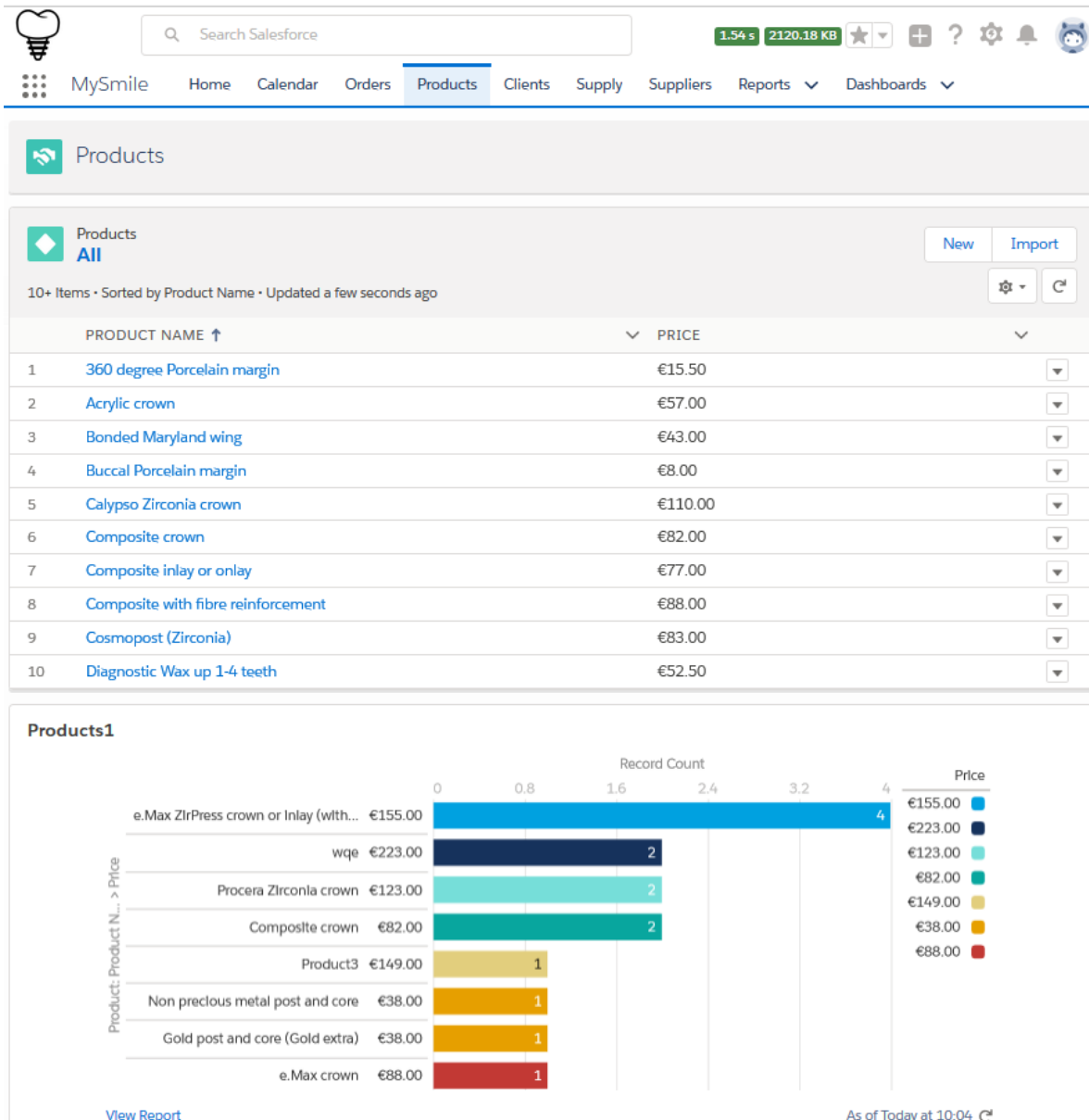
	FIRSTN...	SURNAME	EMAIL	PHONE NUMBER
1	Dr. ena	ena	ena@gmail.com	121 21212
2	Dr. bla	bla		
3	DRneki	neki		
4	Janez	Janez	janez.janez@gmail.com	29287 27272

To the right of the table is a donut chart titled 'ORDERS - Clients this year' showing the 'Sum of Price' for each client. The total sum of prices is 624. The chart is divided into four segments corresponding to the clients: Dr. bla (blue), Dr. ena (dark blue), DRneki (light blue), and Janez (teal). A legend on the right identifies the clients by their first names.

Slika 11: Stran naročnikov

- **Produkt**

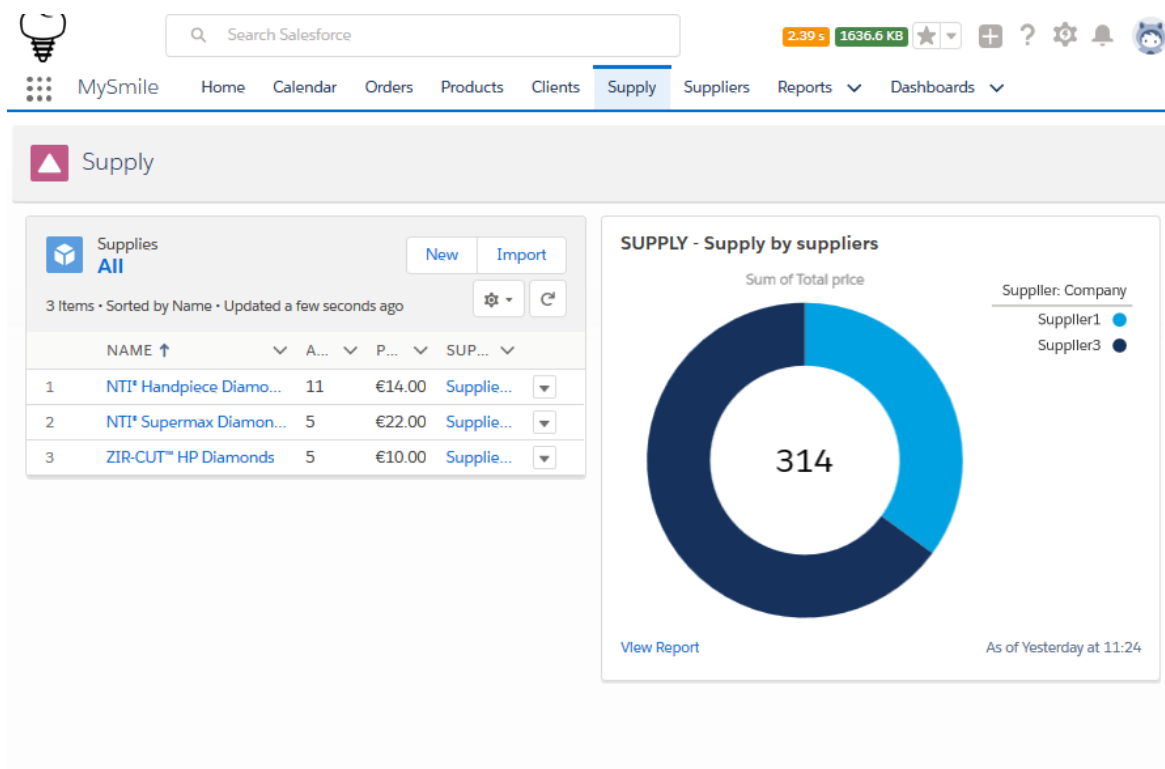
V objektu produkt so vneseni vsi produkti, ki so na voljo za izdelavo v zobotehničnem laboratoriju. Produkt vsebuje ime, opis in ceno produkta, ki v naročilu določa skupno ceno glede na podano količino produktov. Na strani produktov imamo grafični prikaz števila izdelanih produktov ter njihove cene. Prikazuje nam tudi največkrat naročeni izdelek. V zobotehničnem laboratoriju nam to pomaga pri dobavi zaloge ter za pomoč pri poslovanju, saj nam ne sporoča samo, kateri produkt izdelamo največkrat, temveč tudi katerih produktov nam zobozdravniki ne naročajo. Prikaz nam pomaga pri raziskovanju vzrokov, zakaj nam nekaterih produktov zobozdravniki ne naročajo oz. zakaj tako malo. Grafikone v aplikaciji prikazujemo s pomočjo standardnega objekta poročil, ki je opisan v nadaljevanju in nam na podlagi izdelanih poročil omogoča zgraditi grafikon. Prikaz strani produktov je na sliki 9.



Slika 12: Stran produktov

- **Zaloga**

V objektu zaloga imamo podano zalogo v laboratoriju. Izdelana aplikacija nam nudi informativne in statistične podatke. Pri zalogi lahko gledamo datume dobave in znesek, ki smo ga odšteli za določeno zalogo. Na strani je statistični prikaz, pri katerih dobaviteljih naročamo več zaloge. Slika 10 prikazuje stran zaloge v aplikaciji.



Slika 13: Stran zaloge

- **Dobavitelj**

V objektu dobavitelj imamo zapise vseh dobaviteljev. Stran dobaviteljev nam pomaga pri iskanju dobavitelja pri naročanju zaloge. V aplikaciji imamo shranjen naziv, naslov, elektronski naslov, spletno stran in telefonsko številko dobavitelja. Prikaz strani z dobavitelji je na sliki 11.

Suppliers

3 Items · Sorted by Company · Updated a few seconds ago

	COMPANY ↑	EMAIL	WEBPAGE	PHONE	
1	Supplier1	supplier1@gmail.com	www.suplier.com	040 657 2544	▼
2	Supplier2	supplier2@gmail.com	www.Supplier2.com	587 545 658	▼
3	Supplier3	supplier3@gmail.com	www.Supplier3.com	04 15 78 36	▼

SUPPLY - Supply by suppliers

Sum of Total price

Supplier: Company

- Supplier1 ●
- Supplier3 ●

314

[View Report](#) As of Yesterday at 11:24 ↻

Slika 14: Stran dobaviteljev

- **Naročilo**

Naročilo je glavni objekt pri aplikaciji. Pri izdelavi obrazca za vnos novega naročila smo izhajali iz obrazca, ki se uporablja v zobotehničnem laboratoriju. Obrazec je prikazan na sliki 12.

- **ime pacienta;**
- **priimek pacienta;**
- **starost pacienta;**
- **barva zob** – podamo barvo zob pacienta, izberemo jo na podlagi barvnega ključa, ki je prikazan na sliki 13;
- **zgornji zobje** – označimo, za katere zobe na zgornjem loku se izdelata produkt;
- **spodnji zobje** – označimo, za katere zobe na spodnjem loku se izdelata produkt;
- označimo, če izdelek pacient naroči preko **zavarovanja** ali je **samoplačnik**;
- **dodatno** označimo, kar smo dobili dodatno zraven naročila.



Slika 16: Barvni ključ za izbiro barve zob [30]

Pod objektom naročila zaposleni vidijo vsa svoja nedokončana naročila. S pomočjo ustvarjenih pogledov (angl. Views) vidijo tudi svoja naročila, ki imajo podano samo prvo ali samo drugo fazo izdelave. Pogledi so ustvarjeni s pomočjo filtrov, ki jih lahko poljubno dodajamo in na koncu podamo logične operacije med njimi [2].

Pri naročilih imamo grafikon, ki nam prikazuje, koliko dela glede na vrednost naročil je opravil vsak od zaposlenih. Več grafikonov pa imamo na voljo pri poročilih in preglednih ploščah. Posamezno naročilo lahko odpremo in pregledamo vse vnesene podatke. Dodatno vidimo polje s ceno, ki nam izračuna končno ceno glede na naročeni izdelek in velikost naročila. Za prikaz cene smo uporabili polje s podano formulo (angl. Formula Field). Polja s podano formulo lahko uporabljamo za različen prikaz podatkov, lahko jih uporabimo kot prikaz seštevka, odstotka, operacije z datumi, povezave in še mnoge druge operacije. Formulo podamo v urejevalniku formule, in sicer preko enostavnega in naprednega načina [46]. Prikaz strani z naročili je na sliki 14.

The screenshot displays the Salesforce interface for the 'Orders' section. At the top, there is a search bar and navigation tabs for 'MySmile', 'Home', 'Calendar', 'Orders', 'Products', 'Clients', 'Supply', 'Suppliers', 'Reports', and 'Dashboards'. The main content area is divided into three sections:

- My uncompleted orders:** A table with 4 items, sorted by Employee. The columns are O..., C..., PRODUCT, N..., PRICE, FI..., and EMPLO... The items are:

O...	C...	PRODUCT	N...	PRICE	FI...	EMPLO...
1	Novo	Janez wqe	112	€24,976...		Katarina Rov...
2	test1	Janez Product3	2	€298.00		Katarina Rov...
3	R	e.Max ZirPress crown or inl...	2	€310.00		Katarina Rov...
4	Lojzek	Janez e.Max ZirPress crown or inl...	1	€155.00		Katarina Rov...
- First phase:** A table with 1 item, sorted by Order Name. The item is:

ORDER NAME
1 Novo
- Second phase:** A table with 0 items, sorted by Order Name. The message 'No items to display.' is shown.

On the right side, there is a bar chart titled 'ORDERS - orders by employees'. The Y-axis is 'Sum of Price' (0 to 300) and the X-axis is 'Employee: Full Name'. The chart shows two bars: Katarina Rowan with a value of 223 and Luka Ivanšek with a value of 278. Below the chart, it says 'View Report' and 'As of Yesterday at 11:16'.

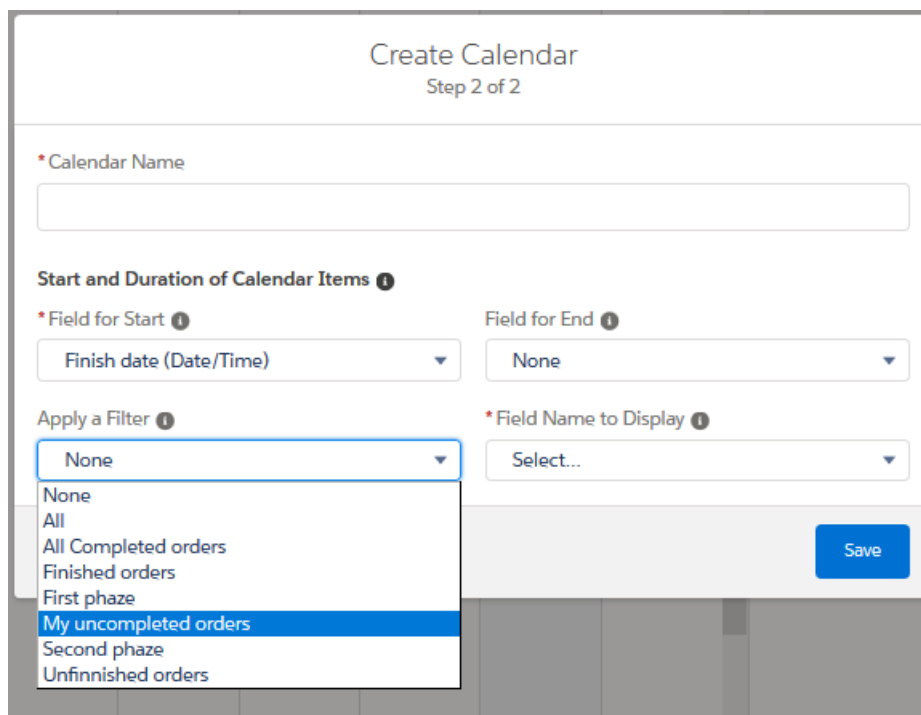
Slika 17: Stran z naročili

5.3.2 Uporabljeni standardni objekti

Standardni objekti so vključeni v platformi Salesforce in so pogosto uporabljeni v podjetjih. V izdelani aplikaciji uporabljamo sledeče standardne objekte: koledar, poročila in pregledne plošče.

- **Koledar**

Koledar nam prikazuje naročila. Vsak uporabnik lahko ustvari prikaz dogodkov v koledarju iz standardnega objekta ali objekta po meri, kot so npr. naročila. Prikaz dogodka določimo z izborom polja z datumom, ki se nam prikaže na koledarju. Ustvarjanje prikaza dogodkov je prikazano na sliki 15.



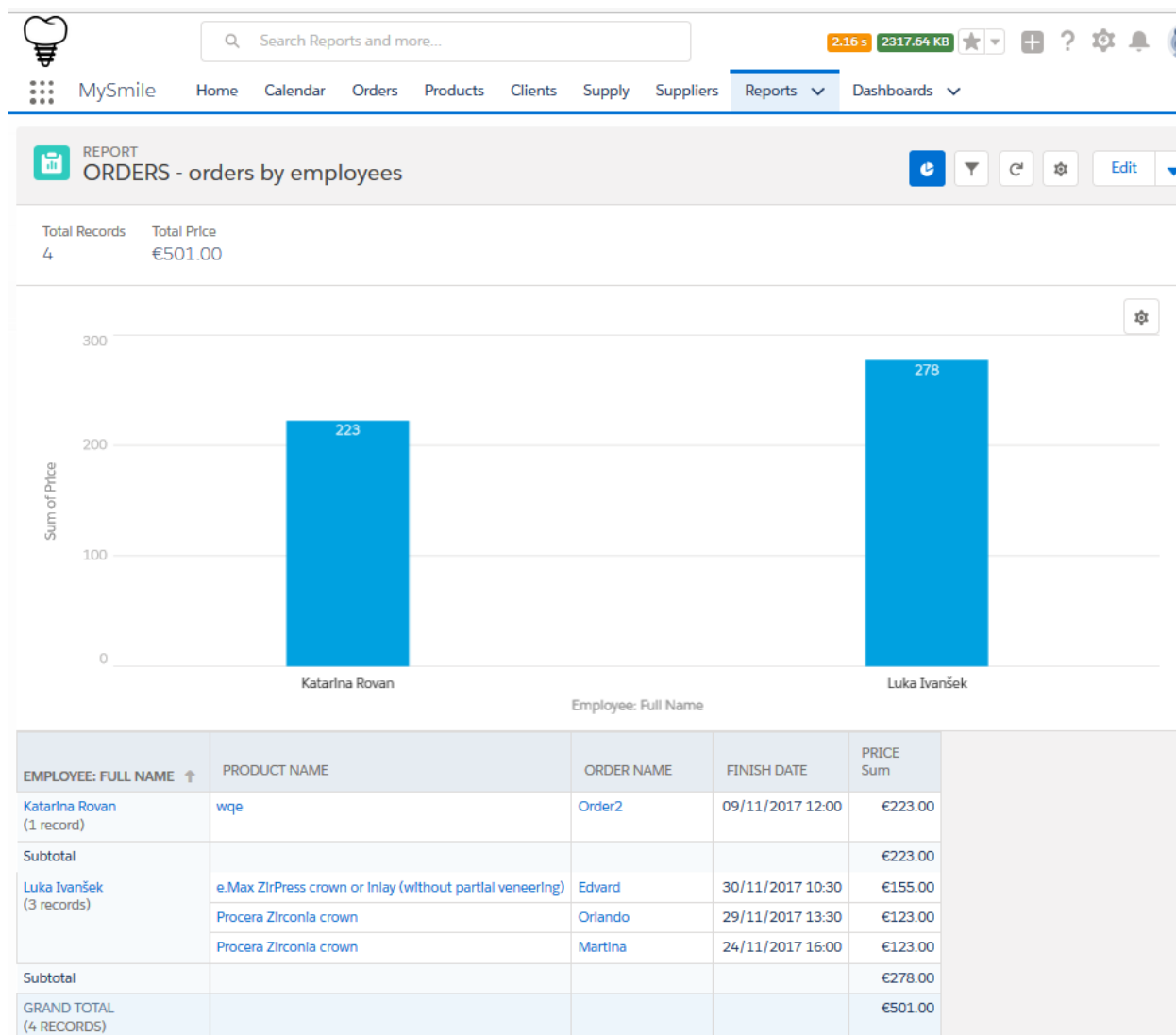
The screenshot shows a web form titled "Create Calendar" at "Step 2 of 2". It contains several input fields and dropdown menus. The "Calendar Name" field is empty. The "Start and Duration of Calendar Items" section has "Field for Start" set to "Finish date (Date/Time)" and "Field for End" set to "None". The "Apply a Filter" dropdown is open, showing a list of filter options: "None", "All", "All Completed orders", "Finished orders", "First phaze", "My uncompleted orders" (highlighted), "Second phaze", and "Unfinished orders". A "Save" button is located at the bottom right of the form.

Slika 18: Ustvarjanje dogodkov za prikaz na koledarju

- **Poročila**

Poročila so uporaben standardni objekt, s katerim izdelujemo vse vrste poročil glede na podane kriterije. Podatke lahko filtriramo, združujemo ali prikazujemo na grafikonu. Poročilo imamo lahko shranjeno v javni, skriti ali deljeni mapi. Določimo lahko dostop uporabnikov do izdelanih poročil.

Vrsta poročila (angl. Report type) je podobna predlogi, ki nam olajša poročanje. Določa nam polja in zapise, ki so na voljo za uporabo pri ustvarjanju poročil. To temelji na razmerju med primarnim objektom s povezanimi objekti [24]. Slika 16 nam prikazuje primer poročila.



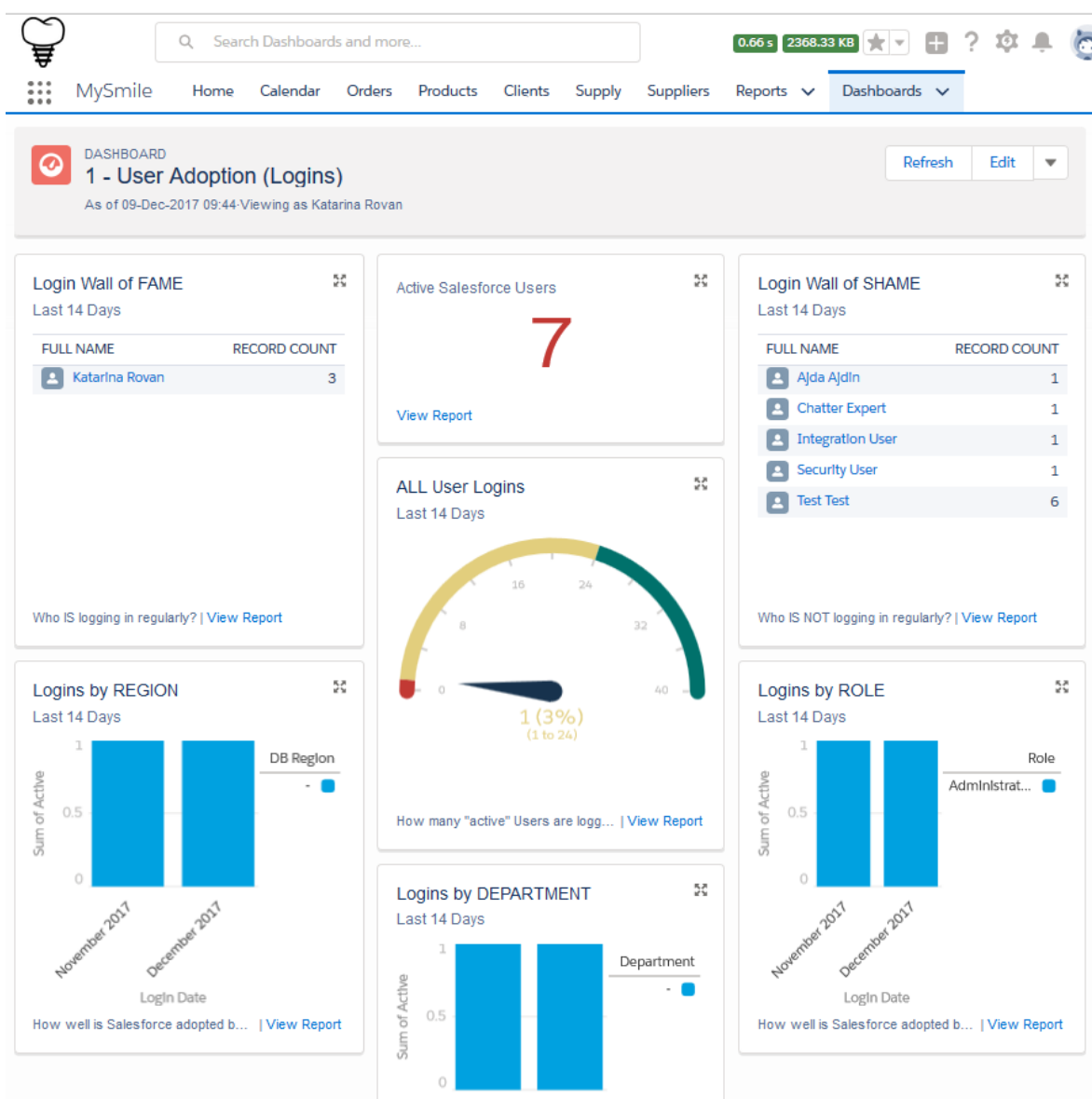
Slika 19: Primer poročila

- **Pregledne plošče**

Pregledne plošče so vizualni prikaz ključnih meritev in trendi zapisov v izdelani aplikaciji. Za vsako komponento pregledne plošče imamo eno poročilo, vendar lahko na isti pregledni plošči uporabimo isto poročilo v več komponentah. Primer uporabe istega poročila v več komponentah je uporaba istega poročila v krožnem diagramu in stolpcnem grafikonu, kjer je komponenta grafikon. Več sestavnih elementov pregledne plošče je lahko prikazanih skupaj na eni strani in omogoča učinkovit prikaz in način za uporabo več poročil s skupno temo, kot so uspešnost prodaje in podpora strankam.

Kot poročila so tudi pregledne plošče shranjene v mapah, kjer določimo dostop uporabnikom. Za ogled pregledne plošče potrebujemo tudi dostop do osnovnih poročil.

Vsaka pregledna plošča ima definiranega uporabnika in varnostne nastavitve tega uporabnika, ki določijo, kateri podatki se bodo prikazovali. Če je uporabnik specifično določen v nastavitvah varnosti, potem vsi uporabniki, ki imajo dostop do njegove pregledne plošče, vidijo podatke, ki temeljijo na njegovih varnostnih nastavitvah, torej neodvisno od svojih nastavitvev, zato moramo biti previdni, komu omogočimo dostop do pregledne plošče. Dinamične pregledne plošče pa določajo definiranega uporabnika kot prijavljenega uporabnika. Na ta način vsak uporabnik vidi pregledno ploščo glede na svojo raven dostopa [24]. Primer pregledne plošče je prikazan na sliki 17.



Slika 20: Primer pregledne plošče

5.4 Mobilna aplikacija

Platforma Salesforce omogoča uporabo izdelane aplikacije preko mobilne aplikacije Salesforce, ki jo dobimo na GooglePlay. Delo v zobotehničnem laboratoriju velikokrat poteka na terenu, pri zobozdravnikih, kjer zobni tehnik prevzame naročilo. Za prevzeta naročila na terenu nam pomaga mobilna aplikacija, saj imamo možnost vnosa novega naročila na mestu prevzema.

5.5 Stroški razvoja aplikacije

Aplikacijo za zobotehnični laboratorij smo razvili na izdaji za razvijalce, ki je brezplačna kopija okolja podjetniške izdaje, s katero nam Salesforce omogoča brezplačni takojšnji razvoj in testiranje izdelane aplikacije. Platforma Salesforce ima mesečna plačila za vsakega uporabnika. Cena osnovne licence za dostop je petindvajset dolarjev na mesec. Za izdelano aplikacijo bi potrebovali osnovno licenco, torej vsaj petinsedemdeset dolarjev na mesec za tri uporabnike aplikacije. Cene se razlikujejo glede na ponujene storitve, kot so omejitve števila objektov po meri, ponudba peskovnikov, ki nam omogočajo nadzorovano okolje za razvoj in preizkus aplikacij, itd. Najpopularnejši paket stane sto dolarjev na mesec na uporabnika, ponujajo pa tudi večji paket za štiri tisoč dolarjev na mesec za podjetje.

Poglavje 6 Sklepne ugotovitve

Z diplomsko nalogo smo raziskali področje računalništva v oblaku, analizirali model platforme kot storitev in zgradili aplikacijo s platformo Salesforce. Aplikacija je razvita z uporabo modela PaaS, ki nam nudi prilagoditve in hitre razširitve brez začetnih stroškov.

Pred pričetkom dela smo ugotovili, kako zobotehnični laboratorij posluje in kakšno aplikacijo potrebuje. Pri izgradnji aplikacije smo si pomagali z nasveti zaposlenega v zobotehničnem laboratoriju.

6.1 Težave

Med gradnjo aplikacije smo naleteli na nekaj težav. Prve težave smo imeli pri razumevanju platforme Salesforce in način, kako platformo uporabljamo za izdelavo aplikacije. Razvoj na Salesforce lahko poteka na Salesforce Classic ali Lightning Experience. Lightning Experience je razvijalno okolje na platformi Salesforce in je sodobnejši in vizualno privlačnejši, vendar nima podprtih vseh Salesforce Classic značilnosti, kot so ustvarjanje novih map za poročila. Začetni razvoj smo začeli na Salesforce Classic, vendar smo zaradi novosti in predvsem zaradi vizualnih lastnosti aplikacijo zgradili na Lightning Experience. Pri razvoju nam je zelo pomagal Trailhead, kjer smo lahko razširili znanje o določenih značilnostih platforme Salesforce in jih implementirali v izdelano aplikacijo.

Platforma Salesforce nudi storitve po določenih cenah z mesečnim plačilom. Storitve niso ravno poceni, torej se manjšim zobotehničnim laboratorijem platforma ne izplača.

Težavo predstavlja tudi dejstvo, da so v Sloveniji po večini manjši zobotehnični laboratoriji, ki so vajeni delo beležiti v pisni obliki in nimajo želje trošiti denarja za aplikacijo.

6.2 Možne nadgradnje

Trenutna aplikacija je dovolj razvita, da je uporabna v zobotehničnem laboratoriju, vendar bi za njeno boljšo uporabnost potrebovali še dodatne nadgradnje. Za boljšo uporabnost bi morali popraviti Salesforce koledar ali pa zgraditi novega, ki bi prikazoval več informacij o naročilu

in bi imel možnost dodajanja naročila. Dodatno bi bilo treba izdelati stran, ki bi nam sestavila račun za naročnika, ter možnost tiskanja računov.

Poglavje 7 Literatura

- [1] About Salesforce Mobile Apps. Dosegljivo: https://developer.salesforce.com/docs/atlas.en-us.mobile_sdk.meta/mobile_sdk/preface_arch.htm [Dostopano 1.12.2017].
- [2] Add Filter Logic. Dosegljivo: https://help.salesforce.com/articleView?id=filter_logic.htm&type=0 [Dostopano 2.12.2017].
- [3] Amazon Web Services. Dosegljivo: <https://aws.amazon.com/> [Dostopano 3.12.2017].
- [4] An Introduction to Cloud Computing Concepts. Dosegljivo: <https://pdfs.semanticscholar.org/9079/25cc2c5be75bebc6d379846b18c0193d0cae.pdf> [Dostopano 18.12.2017].
- [5] An Overview of Force.com Security. Dosegljivo: https://developer.salesforce.com/page/An_Overview_of_Force.com_Security [Dostopano 8.12.2017].
- [6] Apex Governor Limits. Dosegljivo: https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_apexgov.htm [Dostopano 18.12.2017].
- [7] The Chartered Institute for IT BCS, *Cloud Computing, Moving IT out of the office.*: BISL, 2012.
- [8] Benefits And Drawbacks Of PaaS. Dosegljivo: <https://www.getcloudservices.com/blog/benefits-and-drawbacks-of-paas/> [Dostopano 9.12.2017].
- [9] Best PaaS Review. Dosegljivo: <http://www.toptenreviews.com/services/articles/best-paas-review/> [Dostopano 1.12.2017].
- [10] chatter. Dosegljivo: <https://www.salesforce.com/products/chatter/overview/> [Dostopano 2.1.2018].

- [11] Cloud Computing. Dosegljivo: https://en.wikipedia.org/wiki/Cloud_computing [Dostopano 8.12.2017].
- [12] Cloud Computing Information Security and. Dosegljivo: <https://www.ict.govt.nz/assets/ICT-System-Assurance/Cloud-Computing-Information-Security-and-Privacy-Considerations-FINAL2.pdf> [Dostopano 16.12.2017].
- [13] Cloud Deployment Models. Dosegljivo: <http://3.bp.blogspot.com/-X4IUx4udmIc/VFEfuhXVqPI/AAAAAAAAAOGY/oQFIPkbFTkU/s1600/CloudDeploymentModels.jpg> [Dostopano 3.12.2017].
- [14] Code school. Dosegljivo: <https://www.codeschool.com/> [Dostopano 10.12.2017].
- [15] Codecademy. Dosegljivo: <https://www.codecademy.com/> [Dostopano 10.12.2017].
- [16] Compare Lightning Experience and Salesforce Classic. Dosegljivo: https://help.salesforce.com/articleView?id=lex_aloha_comparison.htm&type=0 [Dostopano 18.12.2017].
- [17] Developer Console. Dosegljivo: https://developer.salesforce.com/page/Developer_Console [Dostopano 20.11.2017].
- [18] Jeff Douglas and Wes Nolte, *Salesforce Handbook*. USA, 2010.
- [19] Force.com IDE. Dosegljivo: https://developer.salesforce.com/page/Force.com_IDE [Dostopano 20.11.2017].
- [20] Borko Furht, *Handbook of cloud computing*.: Springer.
- [21] Gamaleldin and Ahmed Mohamed, "An Introduction to Cloud Computing Concepts," *Practical Steps for Using Amazon EC2 IaaS Technology*, pp. 4-6, 2013.
- [22] Google App Engine. Dosegljivo: https://cloud.google.com/appengine/?utm_source=google&utm_medium=cpc&utm_campaign=emea-emea-all-en-dr-skws-all-all-trial-e-gcp-1002258&utm_content=text-ad-none-any-DEV_c-CRE_167351632540-ADGP_SKWS+%7C+EXA+%7E+M%3A1_EMEA_EN_Compute_App+Engine_gae%23SOR-KWI [Dostopano 3.12.2017].

- [23] IdeaExchange. Dosegljivo: <https://success.salesforce.com/ideaSearch> [Dostopano 1.12.2017].
- [24] Introduction to Reports and Dashboards. Dosegljivo: https://trailhead.salesforce.com/en/modules/reports_dashboards/units/reports_dashboards_overview [Dostopano 20.11.2017].
- [25] Matt Kaufman and Michael Wicherski, *Learning Apex Programming*. Birmingham: Packt Publishing Ltd., 2015.
- [26] Logotip AWS. Dosegljivo: <https://media.licdn.com/mpr/mpr/AAEAAQAAAAAAAAAAeRAAAAJDRhMzI2NmY5LWE0NGUtNGEyMS1iNTNiLWRhMmJmOGNiNGMwZQ.png> [Dostopano 8.12.2017].
- [27] Logotip GAE. Dosegljivo: <https://deciphertools.com/blog/img/google-app-engine-logo.jpg> [Dostopano 8.12.2017].
- [28] OAuth 2.0. Dosegljivo: <https://oauth.net/2/> [Dostopano 3.12.2017].
- [29] Jason Ouellette, *Development with the Force. com platform: building business applications in the cloud.*: Addison-Wesley Professional, 2013.
- [30] Paleta barv. Dosegljivo: <https://ae01.alicdn.com/kf/HTB1c7.2RFXXXcXXFXXq6xXFXXXk/Dental-Materiales-Dentales-Material-16-Colores-Dientes-Gu-a-de-colores-Muestras-de-Color.jpg> [Dostopano 8.12.2017].
- [31] Platform as a Service Explained. Dosegljivo: <https://www.salesforce.com/ap/learning-centre/tech/paas/> [Dostopano 3.12.2017].
- [32] Prijava. Dosegljivo: <https://login.salesforce.com> [Dostopano 1.12.2017].
- [33] Salesforce logo. Dosegljivo: <https://developer.salesforce.com/resources2/platform-resources/platform.png> [Dostopano 18.12.2017].
- [34] Kabe Siddhesh, *Force. com Developer Certification Handbook (DEV401)*.: Packt Publishing Ltd, 2012.
- [35] Kabe Siddhesh, *Salesforce Platform App Builder Certification Handbook*.: Packt Publishing Ltd, 2016.

- [36] Skupna infrastruktura. Dosegljivo: <https://res.cloudinary.com/hy4kyit2a/image/upload/doc/trailhead/en-us4f32f70d1ea71f7f0aedab004cafaf62.png> [Dostopano 18.12.2017].
- [37] SOAP API Developer Guide. Dosegljivo: https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_quickstart_intro.htm [Dostopano 3.12.2017].
- [38] Dan Sullivan, *The Definitive Guide to Cloud Computing.*: Realtime Nexus Publishers, 2009.
- [39] Trailhead. Dosegljivo: <https://developer.salesforce.com/blogs/tech-pubs/2015/05/trailhead-new-approach-learning-salesforce.html> [Dostopano 10.12.2017].
- [40] Trailhead site. Dosegljivo: <https://trailhead.salesforce.com/> [Dostopano 10.12.2017].
- [41] Udacity. Dosegljivo: <https://www.udacity.com/> [Dostopano 10.12.2017].
- [42] Understand Custom & Standard Objects. Dosegljivo: https://trailhead.salesforce.com/en/modules/data_modeling/units/objects_intro [Dostopano 3.12.2017].
- [43] Understanding Lightning Experience and Salesforce Classic. Dosegljivo: https://trailhead.salesforce.com/en/modules/lex_migration_introduction/units/lex_migration_introduction_rightforme [Dostopano 3.12.2017].
- [44] Urejanje v vrstici. Dosegljivo: https://2.bp.blogspot.com/-q5B-ywyMjUU/WdLHVMIIdsvI/AAAAAAAAAHLI/sMXIRjyKnoovtDkeLILSq0BY_gsCytIwCLcBGAs/s1600/inline%2Bedit.png [Dostopano 29.1.2018].
- [45] Urejevalnik objekta. Dosegljivo: <http://cloudcoach.com/wp-content/uploads/2015/12/nv8.jpg> [Dostopano 29.1.2018].
- [46] Use Formula Fields. Dosegljivo: https://trailhead.salesforce.com/en/modules/point_click_business_logic/units/formula_fields [Dostopano 20.11.2017].
- [47] Visual Development – When to Click Instead of Write Code. Dosegljivo: <https://developer.salesforce.com/blogs/engineering/2014/12/forcedotcom-declarative-development.html> [Dostopano 10.12.2017].

[48] Visual Studio. Dosegljivo: <https://www.visualstudio.com/> [Dostopano 8.12.2017].

[49] visual-uaas. Dosegljivo:

http://lavancom.com/portal/images/magazine/lavan_magazine/cloud_computing/visual-uaas.png [Dostopano 3.12.2017].