

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
ELEKTROTEHNIČKI FAKULTET**

**Sveučilišni studij**

**SUSTAV ZA PLANIRANJE SASTANAKA ZA GOOGLE  
APP ENGINE OBLAK RAČUNALA**

**Diplomski rad**

**Mirela Čorak**

**Osijek, 2012.**

# Sadržaj

<b>1. UVOD .....</b>	<b>1</b>
<b>2. OBLAK RAČUNALA .....</b>	<b>2</b>
2.1 SVOJSTVA OBLAKA RAČUNALA.....	4
2.2 USLUGE SUSTAVA OBLAKA RAČUNALA .....	6
2.3 PRIMJER POSLUŽIVANJA IZ OBLAKA RAČUNALA .....	8
<b>3. GOOGLE APP ENGINE .....</b>	<b>10</b>
3.1 APP ENGINE ZA JAVU.....	11
3.1.1 <i>Trajna pohrana podataka</i> .....	12
3.1.2 <i>Programska sučelja</i> .....	14
<b>4. MODEL PLANIRANJA SASTANAKA.....</b>	<b>17</b>
4.1 FORMALIZACIJA PROBLEMA.....	17
4.1.1 <i>Dodatni parametri modela</i> .....	22
4.1.2 <i>Daljnje razmatranje</i> .....	30
<b>5. PRIMJENSKI PROGRAM .....</b>	<b>32</b>
5.1 OPIS PRIMJENSKOG PROGRAMA.....	32
5.2 KORIŠTENJE PROGRAMA .....	36
5.3 ANALIZA RADA SUSTAVA.....	44
5.3.1 <i>Primjer uporabe modela</i> .....	45
<b>6. ZAKLJUČAK .....</b>	<b>50</b>
LITERATURA .....	51
SAŽETAK .....	52
ABSTRACT.....	53
ŽIVOTOPIS .....	54
PRILOZI (NA CD-U) .....	55

# 1. Uvod

Internet programi suočavaju se s problemom posluživanja više korisnika nego što je moguće korištenjem resursa jednog poslužitelja. Računarstvo u oblaku omogućuje apstrahiranje arhitekture računala i izvršavanje programa na virtualnim sustavima koji se sastoje od mnogo različitih fizičkih poslužitelja. Takvi sustavi danas su sve popularniji, te većina vodećih informatičkih korporacija razvija vlastite sustave za računarstvo u oblaku.

Cilj rada je opisati mogućnosti Google App Engine platformske usluge oblaka računala, te razraditi model planiranja sastanaka. Planiranje sastanaka je društvena aktivnost koja često zahtjeva znatan napor pri organizaciji. U radu je potrebno detaljno razraditi model planiranja sastanaka i analizirati različite parametre modela. Model je potrebno implementirati u primjenski program koji olakšava dogovaranje vremena održavanja sastanaka. Primjenski program treba postaviti na Google App Engine oblak računala.

Upoznavanje s oblakom računala dano je u drugom poglavlju. Naveden je primjer posluživanja iz oblaka, opisana su svojstva oblaka računala i usluge sustava oblaka računala. Treće poglavlje opisuje Google App Engine oblak računala. Opisane su specifičnosti trajnog pohranjivanja podataka i korisnija programska sučelja sustava. Model planiranja sastanaka opisan je četvrtom poglavlju. Detaljno je formaliziran problem koji se rješava, te su navedeni i analizirani dodatni parametri. U petom poglavlju opisan je implementirani primjenski program. Objasnjeno je korištenje programa, te je navedena analiza. Šesto poglavlje donosi zaključak.

## 2. Oblak računala

Postoje različite definicije računarstva u oblaku (engl. *Cloud Computing*) i svaka od njih naglašava jedan aspekt. Kao što je objašnjeno u [1], osnovna ideja računarstva u oblaku je da se računalni resurs može iznajmiti po potrebi. Pod računalnim resursima misli se na različite stvari kao što su: čitava računala, usluge, aplikacije itd. Korisnik plaća samo ono što je koristio. U trenutku kada trebamo više računalnih resursa, zbog pojave vršnog opterećenja, unajmimo ih bez potrebe za interakcijom s osobljem i bez dodatnih ugovora, a otpustimo ih kada potrebe za njima nestanu – te nas ne brine hoće li skupo plaćena oprema biti iskorištena kao kada sami izgrađujemo sustav. Time se ulaganje u kapitalnu infrastrukturnu opremu tj. poslužitelje, mrežne i strujne kapacitete i posebne poslužiteljske prostorije pretvara u operativne troškove jer se plaća prema korištenju.

Prema [7] sustav oblaka računala može se definirati kao:

- sustav u kojemu je moguće iznajmljivanje jednog ili više poslužitelja, te pokretanje različitih programa na njima,
- sustav u kojemu je moguće iznajmljivanje virtualnog poslužitelja na kojemu korisnici mogu pohranjivati podatke i po volji im pristupati,
- sustav u kojemu je moguće pohranjivanje i osiguravanje ogromnih količina podataka kojima mogu pristupiti samo ovlašteni programi ili korisnici,
- sustav u kojemu je moguće korištenje programa koji se nalazi na internetu i pohranjuje te štiti podatke za vrijeme pružanja usluge,
- sustav u kojemu je moguće koristiti virtualne poslužitelje za pohranu podataka kako bi se na njima čuvali programi, poslovni i osobni podaci,
- sustav u kojemu je moguće korištenje mnoštva internet programa, te je uz pomoć njih moguće integrirati fotografije, karte, GPS informacije i druge korisne stvari.

Ukratko, može se reći da je oblak računala sustav uz pomoć kojeg se može pristupiti programima i ostalim podacima koristeći internet. Slika 2.1 prikazuje da su korisniku programi i ostali podaci dostupni sa svakog računala s kojeg korisnik ima pristup internetu.



**Sl. 2.1. Pristup sustavu oblaka računala**

Ako se neka organizacija i tvrtka odluči napraviti svoj podatkovni centar potrebno je imati stručne inženjere koji znaju posložiti (instaliranje i konfiguriranje) i kasnije održavati takav centar. Osim toga je potrebno osigurati: mrežni pristup dovoljnog kapaciteta, opskrbu električne energije (npr. 20.000 računala troši 500KW električne energije), hlađenje prostora u kojem su računala, fizičku sigurnost, barem dvije lokacije zbog pouzdanosti, ... Cijena jednog velikog podatkovnog centra iznosi od \$300.000.000 do \$2.000.000.000, a prostor na kojem su napravljeni se kreće od  $30.000\text{ m}^2$  do  $300.000\text{ m}^2$ . Velike tvrtke kao što su Google, Microsoft, Amazon imaju od 5 do 30 takvih centara diljem svijeta. Poslužitelji u tim centrima se koriste od 3 do 5 godina, a njihova se isplativost postiže nakon 18 mjeseci, prvenstveno zbog potrošnje energije. Za takav centar se biraju lokacije koje imaju pristup brzom internetu (blizu magistralnih veza), jeftinu električnu energiju i mogućnost korištenja ekoloških izvora energije (vjetroelektrane, solarni paneli, ...).

U nastavku se opisuju pogodnosti koje oblaka računala pruža svojim korisnicima, te su pojašnjene neke prednosti i nedostaci.

## 2.1 Svojstva oblaka računala

Kod računarstva u oblaku korisniku više nisu potrebni fizički hardveri, već se koriste hardveri koji se nalaze u oblaku. Oblaku računala korisnik može pristupiti s vlastitog računala, te nije važno gdje se korisnik nalazi jer uvijek može pristupiti poslužitelju oblaka i kontrolirati ga. U oblaku je moguće pohranjivati i čuvati podatke i programe, te im pristupati i mijenjati ih u bilo kojem trenutku, gdje god se nalazili. Ključno svojstvo oblaka računala je nepovezanost korisnika s hardverom, tj. korisnik se ne mora brinuti za hardver, nego samo za svoje programe. Korisnicima nije važno gdje se nalaze fizički poslužitelji na kojima se pohranjuju korisnički podaci, važno im je kako usluga funkcioniра, te da je usluga sigurna, stabilna i jednostavna za korištenje.

Postoje mnogi razlozi zašto je računarstvo u oblaku sve popularnije:

- **Skalabilnost** – jednostavno je povećavati ili smanjivati broj poslužitelja potreban za posluživanje korisnika. Automatski se proširuju mrežni i računalni kapaciteti ovisno o opterećenju. Npr. ako internet aplikacija koju smo postavili u oblak zahtjeva više računalnog kapaciteta u nekom periodu, sustav će to detektirati i povećati kapacitete u tom periodu ili smanjiti kapacitete u nekom drugom periodu.
- **Trenutna pristupačnost** – računalni resursi su pristupačni kad god su potrebni, samo ih treba aktivirati. Sve što korisniku treba odmah je pristupačno u oblaku, a kada se korisnikove potrebe smanje jednostavno se resursi oslobole.
- **Ušteda novca** – korisnik plaća samo ono što koristi, a ne i opremu koja će stajati nezaposlena čekajući da ju korisnik zatreba. Korisnici se ne moraju brinuti o hardveru, te se mogu više posvetiti izgradnji posla. Ne zahtijevaju se kapitalni troškovi za izgradnju infrastrukture – nije potrebno kupovati, instalirati i održavati poslužitelje. Ne zahtjeva se niti vrijeme niti inženjersko znanje za postavljanje sustava. Koristeći oblak računala korisnik si može uštedjeti vrijeme i novac.
- **Pružanje usluge na zahtjev korisnika** (engl. *On-demand self-service*) – korisnik samostalno odabire i pokreće računalne resurse, te samostalno bira vrijeme posluživanja i mrežni prostor za pohranu podataka. Korisnik nema potrebe za interakcijom s djelatnicima pojedinog davatelja usluge. Većina

poslužitelja svoje usluge temelji upravo na pristupu u kojemu korisnici plaćaju usluge u ovisnosti o vremenu i obujmu u kojem ih koriste..

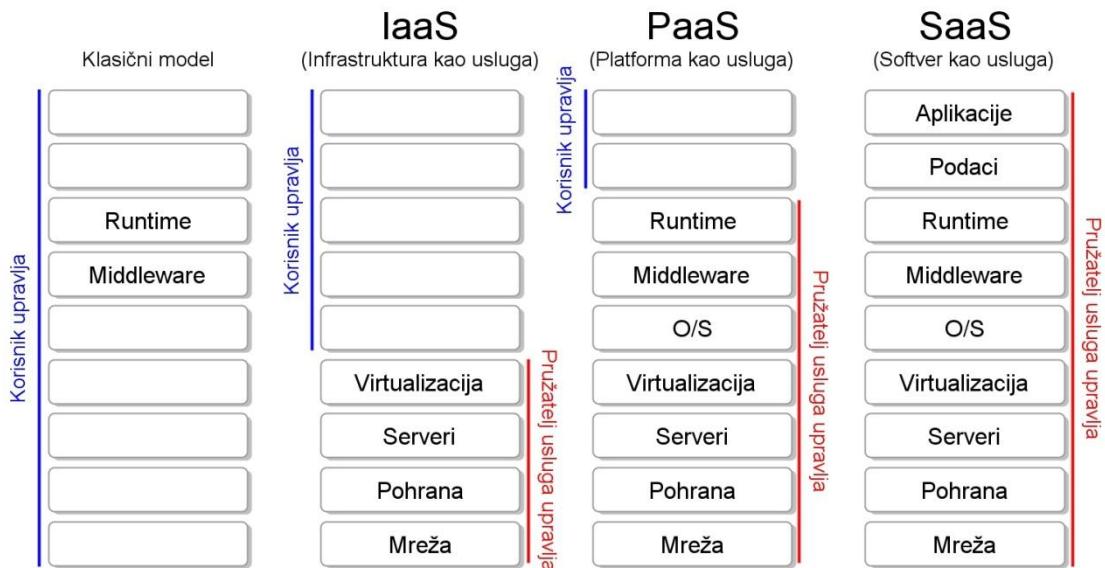
- **Širok mrežni pristup** (engl. *Broad network access*) – sustav je dostupan preko mreže, te mu se pristupa koristeći standardne komunikacijske protokole. Omogućuje se spajanje bilo kojeg uređaja na sustav od bilo kuda.
- **Udruživanje resursa** (engl. *Resource pooling*) – računalni resursi pružatelja usluga spajaju se kako bi poslužili što više korisnika koristeći model s više zakupljenih jedinica (engl. *Multi-Tenant model*), s različitim fizičkim i virtualnim resursima, koji se dinamički zauzimaju i oslobođaju prema zahtjevima korisnika. Korisnik obično ne zna na kojem se točno mjestu nalaze zauzeti resursi, ali ih ipak može odrediti na većoj razini apstrakcije. Pod pojmom resursa smatraju se mrežni prostor, procesori, memorija, mrežna propusnost, te virtualni strojevi.
- **Brza elastičnost** (engl. *Rapid elasticity*) – mogućnosti koje oblak računala nudi korisnicima mogu biti ubrzane i elastično pokrenute, kako bi se po potrebi ostvarilo proporcionalno povećanje ili smanjenje mogućnosti kada više nisu potrebne. Krajnjem korisniku korištene mogućnosti mogu izgledati kao da nemaju ograničenja i mogu se kupiti u bilo kojoj količini u bilo koje vrijeme.
- **Odmjerena usluga** (engl. *Measured service*) – sustavi koji koriste oblake računala automatski provjeravaju i optimiraju uporabu resursa. Uporaba resursa se može pratiti, provjeravati, te se mogu raditi izvješća – pružajući tako transparentan uvid davateljima usluge i korisnicima.
- **Sigurnost** (engl. *Security*) – oblak računala oslobađa pojedinca i organizaciju od troškova i napora instaliranja, održavanja i stalnog unapređivanja programa na svojim osobnim računalima i u svojim podatkovnim centrima. Oblak računala oslanja se na pouzdanu internetsku vezu, te ujedno stvara ovisnost o vanjskim dobavljačima koji mogu povećati zaštitu podataka i smanjiti sigurnosne rizike. U oblaku su ugrađeni sustavi koji omogućuju uravnoteženje opterećenja i visoku dostupnost, a svaki oblak ima i više fizički raspodijeljenih podatkovnih centara koji su povezani brzom mrežom.

Razvoj aplikacija za oblak računala ima i svoje nedostatke:

- Zahtjeva se razvoj aplikacija koje ne pamte međustanja (engl. *stateless*) (kao što HTTP protokol ne pamti stanje, tako stanje ne pamte niti oblaci računala). Najjednostavnije aplikacije često ne pamte stanja, međutim, za razvoj zanimljivijih aplikacija potrebno je na neki način pamtiti stanje komunikacije – što opravdava korištenje baza podataka u oblaku računala.
- Oblak se jako brzo mijenja! Program mora biti spreman na česte promjene, nove pluginove i proširenja funkcionalnosti oblaka.
- Ne postoji mogućnost prilagodbe klijentu kao kada klijent sam upravlja sa svojim fizičkim računalima.
- Privatnost i sigurnost podataka je najveći nedostatak jer klijent ne zna gdje su podaci i na koji način su pohranjeni. Problem mogu uzrokovati lokacije podatkovnih centara jer se lokalni zakoni o sigurnosti podataka razlikuju. Npr. utjecaj politike na filtriranje podataka (poznat je slučaj u kojem je Google imao podatkovni centar u Kini, pa je pristup tim podacima bio filtriran).

## 2.2 Usluge sustava oblaka računala

Postoje tri različite usluge oblaka računala (prikazane na slici 2.2) koje omogućuju korisnicima pokretanje programa i pohranjivanje podataka. Svaka od tih usluga pruža različitu razinu fleksibilnosti i kontrole korisnika. Na korisniku je da izabere onu koja najviše odgovara njegovim zahtjevima.



Sl. 2.2. Usluge sustava oblaka računala

- **SaaS** (engl. *Software as a Service*) – je usluga oblaka računala koja omogućuje korisnicima jednostavan pristup programima putem interneta. SaaS usluga omogućuje svojim korisnicima da koriste programe poslužitelja koji se pokreću na infrastrukturi oblaka. Takvi programi su dostupni različitim klijentskim uređajima pomoću internet preglednika. Prednost SaaS usluge je upravo to što korisnici ne moraju više kupovati programe i instalirati ih na vlastito računalo, nego im mogu pristupiti preko interneta s bilo kojeg uređaja. Korisniku nije dopušteno upravljanje infrastrukturom oblaka, uključujući mrežu, operacijske sustave i pohranu podataka. Korisniku je jedino dopušteno mijenjanje konfiguracijskih postavki programa. Primjeri SaaS-a su: SalesForce CRM, GMail, Google Docs, DeskAway, Impel CRM, Oracle On Demand, SQL Azure, Google Apps, Microsoft Office 365 i drugi.
- **PaaS** (engl. *Platform as a Service*) – je varijacija SaaS strukture koja kao uslugu donosi razvojno okruženje. Korisniku se omogućava da sam kreira vlastite programe koristeći programske jezike i alate koje mu pruža poslužitelj, te da ih postavi na infrastrukturu oblaka. PaaS usluga ima neka ograničenja kojih se korisnik mora pridržavati kako bi razvijen program bio funkcionalan i siguran. Korisniku nije dopušteno upravljanje infrastrukturom oblaka uključujući mrežu, operacijski sustav ili mjesto pohrane podataka – za to se brine poslužitelj.

Korisniku je jedino dopušteno upravljanje razvijenim programima (koji su postavljeni na infrastrukturi oblaka) i omogućeno mu je mijenjanje konfiguracije okruženja poslužitelja. Neki od primjera su: Force.com, GoGrid Cloud Center, Windows Azure Platform, Salesforce.com, Coghead i Google App Engine.

- **IaaS** (engl. *Infrastructure as a Service*) – korisniku kao uslugu pruža mogućnost korištenja računalne infrastrukture. IaaS usluga pruža svojim korisnicima procesore, memoriju, pohranu podataka, umrežavanje i sigurnost. IaaS je virtualni stroj pomoć kojeg korisnik može birati između nekoliko operacijskih sustava (većina korisnika radije bira open source operacijske sustave), količinu hardvera (procesorska snaga) i količinu prostora za pohranu. Prednost IaaS usluge su minimalni početni troškovi i mogućnost korisnika da plaća samo ono što koristi. Kapitalni izdaci se uz pomoć IaaS usluge mogu svesti na minimum, što je vrlo pogodno za mala poduzeća. Primjeri IaaS usluge su: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Eucalyptus i GoGrid.

## 2.3 Primjer posluživanja iz oblaka računala

Pojasnimo kroz primjer kako funkcionira posluživanje u oblaku.

Prepostavimo da neka tvrtka ima internet stranicu koja služi za komunikaciju između korisnika. U početku tu internet stranicu koristi samo nekolicina ljudi. Ako je internet stranica kvalitetna i vrijedna posjeta uskoro će imati sve više i više korisnika. Tvrta će postati uspješnija, ali s uspjehom dolaze i problemi sa skalabilnošću. Velik broj korisnika znači veći broj zahtjeva, a ako oprema za posluživanje ne može izdržati pritisak koji zahtjevi rade nad serverom – usluga će postati sporija ili čak nedostupna. Tradicionalno posluživanje u takvim slučajevima zahtjeva kupnju nove opreme, tj. novih poslužitelja, te njihovo postavljanje. Mnoge tvrtke i danas koriste tradicionalan način posluživanja što zahtjeva dodatne financijske resurse. Potrebno je vrijeme kako bi se postavila dodatna oprema, te se dodatno povećaju i troškovi održavanja. Tvrta mora plaćati poslužitelje neovisno o tome koristi li postavljene računalne resurse ili ne. Napretkom tehnologije danas postoje bolja rješenja za internet posluživanje, a to je računarstvo u oblaku. S oblakom računala korisnici odmah imaju pristup računalnoj snazi koja im je potrebna. Ako internet stranicu posjećuje više ljudi nego je predviđeno,

te je potrebno više računalnih resursa da bi se stranica posluživala, tvrtka može dodati onoliko računalne snage koliko smatra potrebnim. Dodatnu računalnu snagu pruža oblak, dok posjetitelji stranice ne primjećuju nikakve promjene. Kada se potreba za računalnom snagom smanji korisnik jednostavno otpusti poslužitelje koji mu više nisu potrebni. Glavna prednost računarstva u oblaku je upravo to: kada je potrebno više računalne snage, može se dobiti istog trena, a kada ona više nije potrebna otpusti se natrag u oblak.

### 3. Google App Engine

Google App Engine, opisan u [5], je PaaS (engl. *Platform as a Service*) oblak računala koji omogućava korisnicima pokretanje internet aplikacije na Googlevoj infrastrukturi (prvenstveno skalabilni GFS datotečni sustav i Googlova BigTable NoSQL baza podataka). App Engine omogućuje automatizirano skaliranje broja poslužitelja ako se poveća količina zahtjeva za aplikacijom.

Google App Engine omogućava posluživanje internet aplikacija napisanih u programskim jezicima koji se izvršavaju unutar Java, Python ili Go izvršnog okružja (engl. *runtime environment*). Korištenje App Engine oblaka naplaćuje se po potrošnji resursa, a ne po vremenu posluživanja.

Google App Engine osigurava korisničkoj aplikaciji pouzdano izvršavanje pod velikim opterećenjem i s velikim skupovima podataka. App Engine omogućuje:

- Dinamičko posluživanje.
- Trajnu pohranu podataka i upite nad podacima (Google Query Language) s podržanim transakcijama.
- Automatsko skaliranje i raspoređivanje opterećenja.
- API-ji (engl. *Application programming interface*) za autentificiranje korisnika i slanje emailova pomoću Googlovih korisničkih računa.
- Lokalno razvojno okružje koje simulira Google App Engine.
- Redove zadataka (engl. *Task queues*) pomoću kojih se aplikacijski kod može izvršavati i izvan *http* zahtjeva.
- Zakazane zadatke (engl. *Scheduled tasks*) za okidanje događaja u određenim trenucima ili po definiranim periodičnim intervalima.

Korisničke aplikacije se izvršavaju na serveru bez da utječu na aplikacije drugih korisnika. Sigurnosno okružje (engl. *Sandbox*) ograničava pristup operacijskom sustavu na kojem se aplikacija izvršava. Ta ograničenja omogućuju App Enginu da distribuirira zahtjeve za pristup aplikaciji na više različitih servera, te da po potrebi dodaje i uklanja servere ovisno o količini zahtjeva za aplikacijom. Neka od ograničenja koja okružje nameće su:

- Aplikacija može pristupiti drugim računalima na internetu samo pomoću URL adrese i pomoću email servisa; dok druga računala na internetu mogu pristupati aplikaciji samo pomoću HTTP (ili HTTPS) zahtjeva.
- Aplikacija ne može direktno pisati po datotekama operacijskog sustava. Aplikacija može čitati samo one datoteke koje su sastavni dio aplikacije (vlasnik aplikacije ih je postavio na server). Aplikacija mora trajno pohraniti podatke, postaviti podatke u privremenu memoriju (engl. *memcache*) ili koristiti neki drugi dostupni servis, kako bi očuvala podatke između korisničkih zahtjeva (čak se i podaci zapisani u sjednicu (engl. *session*) zapisuju u bazu podataka između dva uzastopna zahtjeva za pristup aplikaciji).
- Aplikacijski kod se izvršava samo kao odgovor na korisničke zahtjeve, redove zadataka i zakazane zadatke. Dretva koja obrađuje zahtjev ne može pokrenuti pod-proces ili izvršavanje koda nakon što se pošalje odgovor na zahtjev.

## 3.1 App Engine za Javu

Aplikaciju koja koristi Java razvojno okružje (engl. *Java Runtime Environment*) moguće je implementirati korištenjem uobičajnih razvojnih alata i API standarda korištenih za razvoj Java Web aplikacija. Java razvojno okružje koristi Javu 6, dok App Engine Java SDK (engl. *Software development kit*) podržava razvoj aplikacija za Javu 5 i Javu 6.

Aplikacija komunicira s razvojnim okružjem pomoću Java Servlet standarda, te može koristiti uobičajne internet tehnologije poput JavaServer Pages (JSP). Korisnik stvara servlet klase, JavaServer Pages, statičke datoteke (*html*, *xml*, *css*, *js*, slike i ostalo), opisnik aplikacije (engl. *deployment descriptor*) i ostale konfiguracijske datoteke u standardnoj WAR strukturi direktorija.

Sigurnosna ograničenja su implementirana direktno u JVM (engl. *Java virtual machine*). Aplikacija može koristiti bilo koji JVM binarni kod (engl. *bytecode*) i svojstvo biblioteke dok god ne prekrši sigurnosno ograničenje. Na primjer, JVM binarni kod koji pokuša otvoriti priključak (engl. *socket*) ili pisati u datoteku uzrokovat će bacanje iznimke (engl. *exception*).

App Engine pruža skalabilne servise koje aplikacija može koristiti kako bi trajno pohranila podatke, dohvatala resurse preko interneta, ili izvršila druge zadatke poput manipuliranja slikom.

### 3.1.1 Trajna pohrana podataka

Korisniku su na raspolaganju dva različita Java standarda za trajno pohranjivanje (engl. *persistent*) Java objekata:

- Java Data Object 2.3 standard (JDO) i
- Java Persistence API 1.0 standard (JPA).

Oba su implementirana pomoću DataNucleus platforme. Također je na raspolaganju i kompletno sučelje niske razine (engl. *low-level API*) za direktni pristup i za implementiranje adaptera za druge standarde. JDO i JPA standardi su lakši za uporabu i transparentni prema drugim implementacijama tih standarda, ali s druge strane zahtijevaju više komunikacije s bazom i manje su otporni na greške.

App Engine skalabilna NoSQL pohrana podataka omogućava:

- Atomske (engl. *atomic*) transakcije – transakcije koje se ili provedu u potpunosti ili se ne provedu (nije moguće napola provesti transakciju).
- Visoku dostupnost paralelnih čitanja i pisanja.
- Čvrstu konzistentnost (engl. *strong consistency*) upita nad predcima – upit nad podacima koji imaju definiranog zajedničkog pretka garantira konzistentnost poretku čitanja i pisanja koji se izvršavaju nad istim pretkom.
- Konačnu konzistentnost (engl. *eventual consistency*) nad ostalim upitim – svi upiti su konzistentni nakon što prođe konačno mnogo vremena.

App Engine kao primarni način pohrane podataka koristi High Replication Datastore (HRD) u kojem se podaci repliciraju (obično 3 kopije) na više podatkovnih centara pomoću sustava koji je temeljen na Paxos algoritmu.

Pohranjeni podaci se spremaju u podatkovne objekte koji se nazivaju *entiteti* (engl. *entities*). Svaki *entitet* ima jedan ili više imenovanih *svojstava* (engl. *properties*) koji pohranjuju neki od dozvoljenih tipova podataka. Na primjer, *svojstvo* može biti string, cijeli broj ili referenca na drugi *entitet*. Svaki *entitet* je određenog *tipa* (engl. *kind*) i ima jedinstveni ključ (engl. *key*) koji ga identificira unutar *tipa*. *Tip* grupira *entitete* kako bi se mogli koristiti u upitim. U usporedbi s klasičnim relacijskim bazama podataka

mogli bi reći da je *tip* sličan pojmu tablice, *entitet* je sličan retku unutar tablice, dok je *svojstvo* neka od vrijednosti unutar tog retka. Ipak, App Engine pohrana podataka je NoSQL te nema smisla tražiti previše sličnosti s klasičnim SQL bazama podataka. Neka od specifičnosti App Engine pohrane podataka su:

- *Entiteti istog tipa* ne moraju imati ista *svojstava*, niti im *svojstva* moraju pohranjivati iste tipove podataka.
- *Svojstvo* može sadržavati više podataka koji mogu biti različitog tipa.
- Svaki *entitet* ima ključ (*tip* nema ključ) koji ga čini jedinstvenim unutar *tipa* kojem pripada. Jedinstveni ključ automatski generira sustav za pohranu podataka.
- Maksimalna veličina *entiteta* je 1MB. Pojedini *entitet* može imati najviše 5000 zapisa u svim indeksima koji ga adresiraju.
- Upit može vratiti jedan ili više *entiteta*, te ih može vratiti sortirane po željenom *svojstvu*.
- Upit može ograničiti broj vraćenih *entiteta* kako bi štedio prostor ili vrijeme izvršavanja.
- *Entitet* se selektira upitom ako ima odgovarajuću vrijednost za svako *svojstvo* nad kojim se provodi filtracija po kriterijima upita.
- Transakcije trebaju biti idempotentne (otporne na višestruko ponovljeno izvršavanje) jer neuspjela transakcija (ukoliko dođe do greške) može u potpunosti završiti zapisivanje, te uzrokovat grešku pri vraćanju informacije aplikaciji (koriste se nesigurni komunikacijski mediji).
- Upiti i dohvati će vidjeti konzistentnu sliku podataka dobivenu u trenutku kada transakcija započne.
- Ako se ne sortira rezultat upita, moguće je da svaki upit vidi drukčiji poređak *entiteta*.
- Brzina pisanja u istu grupu *entiteta* je ograničena na jedno pisanje po sekundi (posljedice ovog ograničenja se mogu ublažiti korištenjem Memcache sučelja).
- JOIN radnje i pod-upiti nisu podržani jer bi usporili izvršavanje upita.
- Vrijeme izvršavanja upita ovisi o veličini vraćenih podataka (tj. veličini rezultata upita), a ne o veličini podataka nad kojim se upit izvršavao. Npr. upit koji vraća sto *entiteta* jednak je dugu izvršavao nad skupom od sto *entiteta* i nad skupom

od sto milijuna *entiteta*. Ovo iznimno svojstvo dovoljno opravdava korištenje NoSQL baze podataka za sve skalabilne aplikacije koje koriste velike skupove podataka.

App Engine može direktno pohranjivati sljedeće tipove podataka:

1. Null vrijednost
2. Cijeli broj
  - Integer
  - Datum i vrijeme
  - Ocjene
3. Boolean vrijednost
4. Byte string (do 500 znakova)
5. Unicode string
  - Tekstualni string (do 500 znakova)
  - Poštanska adresa
  - Telefonski broj
  - Email adrese
  - IM korisničko ime
  - Link
  - Kategorija
6. Decimalni broj
7. Geografska koordinata
8. Google korisnički račun
9. Ključ entiteta
10. Ključ za Blobstore

App Engine također pruža sučelje za dohvaćanje metainformacija o podacima i statistike o pristupu podacima.

### 3.1.2 Programska sučelja

App Engine pruža mnoštvo programskih sučelja:

- Za pohranu podataka, osim trajne pohrane, raspoloživ je i mehanizam pohranjivanja u priručnu memoriju (engl. *cache*); raspoloživ pomoću **Memcache** sučelja. Java sučelje također implementira i JCache standard (JSR 107). App Engine Memcache omogućava brzu i skalabilnu tranzientnu pohranu podataka koja se može koristiti npr. za privremeno pamćenje rezultata dohvata podataka iz trajne memorije. Memcache se može koristiti i za sinkroni i za

asinkroni dohvati podataka. Pojedina pohranjena vrijednost mora biti manja od 1MB.

- **Blobstore API** omogućava aplikaciji posluživanje podatkovnih objekata zvanih *blob* koji mogu biti znatno veći od dopuštene veličine objekata koji se pohranjuju na uobičajen način. *Blobovi* se stvaraju učitavanjem datoteka pomoću HTTP zahtjeva. Blobstore API pohranjuje učitani *blob* te vraća referencu na *blob*. *Blob* se može dohvaćati pomoću reference i posluživati kao odgovor na korisnički zahtjev.
- **Cloud Storage API** omogućava pohranu podataka pomoću RESTful sučelja, te podržava deklarativno filtriranje poslanih datoteka. Cloud Storage API također podržava OAuth 2.0 standard za autorizaciju i autentifikaciju.
- **Images Java API** omogućava manipuliranje slikama pomoću zasebnih servera posvećenih samo obradi slike. Servis može skalirati, rotirati, zrcaliti i izrezivati sliku. Također omogućava uljepšavanje slike pomoću nekoliko različitih algoritama.
- Poput Images Java sučelja, dostupno je **Conversation API** sučelje koje omogućuje pretvorbu dokumenata između sljedećih formata: *html, pdf, gif, jpg, bmp, txt* i *png*. Sučelje se može koristiti sinkrono i asinkrono, te omogućava prepoznavanje teksta sa slike (engl. *optical character recognition*).
- **URL Fetch API** omogućava dohvaćanje različitih resursa dostupnih na internetu pomoću HTTP i HTTPS protokola. Aplikacija može pomoću URL Fetch sučelja slati zahteve drugim internet aplikacijama. URL Fetch sučelje se može koristiti pomoću sučelja niske razine ili korištenjem standardne Javine biblioteke *java.net*.
- **Channel API** omogućava stvaranje trajne konekcije između klijenta i aplikacije; tako da aplikacija može poslati podatke klijentu bez da klijent pokrene komunikaciju (poznato kao Ajax Push ili Comet tehnologije). Channel API je iznimno koristan za aplikacije koje moraju slati korisniku nove podatke u stvarnom vremenu (npr. chat).
- **LogService API** omogućava aplikacijama bilježenje serverskih logova. Logiranje se može provoditi pozivom LogService sučelja ili direktnim ispisivanjem na standardni ispis i standardni ispis za greške (sve ispisano se

logira). Logirane informacije se mogu dohvaćati programski, te u administratorskoj konzoli.

- **Mail Java API** omogućava slanje elektroničke pošte pomoću s Googleovih servera. Osim sučelja niske razine, Mail Java sučelje se može koristiti i pomoću standardne Javine biblioteke *javax.mail*. U aplikaciji je moguće obrađivati primljene mailove tako što se u konfiguracijskim datotekama mapira putanja koja se poziva za primljeni mail (na definiranu putanju se registrira željeni Servlet).
- **Prospective Search** omogućava registriranje većeg broja upita koji se izvršavaju nad velikim skupom protočnih podataka. Korisno za aplikacije koje se bave nadziranjem, filtriranjem ili obavještavanjem.
- **Users Java API** omogućava autentifikaciju korisnika koji imaju Googleov korisnički račun, posjeduju OpenID identifikator, ili imaju račun na vlastitoj Google Apps domeni. Sučelje omogućuje detekciju je li korisnik ulogiran, te može zahtijevati prikaz Googleove forme za logiranje kako bi ga se autentificiralo. Sučelje također omogućuje detektiranje je li prijavljeni korisnik administrator, kako bi mu se omogućio privilegirani pristup aplikaciji. Osim navedenog sučelja dostupno je i **OAuth for Java** sučelje koje omogućava autentifikaciju korisnika koji koriste standardni OAuth protokol.
- **Task Queue API** omogućava izvršavanje programskog koda van korisničkog zahtjeva. Ako aplikacija mora izvršiti nešto u pozadini može koristiti Task Queue kako bi organizirala posao u manje zadatke (engl. *task*). Aplikacija doda zadatke u red za izvršavanje, te se oni izvršavaju brzinom podešenom u odgovarajućoj konfiguracijskoj datoteci. Izvršavanje zadataka se može zakazati na željeni trenutak u budućnosti. Budući da se izvršavanje zadatka pokrene pozivanjem postavljene putanje, zadatak se smatra uspješnim samo ako vrati HTML statusni kod između 200 i 299 (neuspjeli zadaci se pokušaju ponovno pokrenuti). Zadaci bi trebali biti idempotentni budući da se u rijetkim slučajevima može dogoditi da se isti zadatak izvrši više puta. Task Queue se može koristiti i pomoću REST API sučelja.

## 4. Model planiranja sastanaka

Ljudi, kao društvena bića, često organiziraju sastanke kako bi se mogli okupiti u istom trenutku na istom mjestu, te se družili/surađivali/komunicirali. Pritom dogovaranje termina sastanka često predstavlja problem, budući da ljudi zauzimaju svoje raspoloživo vrijeme raznim drugim aktivnostima.

Dogovaranje termina sastanaka može se olakšati korištenjem aplikacije koja olakšava suradnju među ljudima. Računalom podržana suradnja pri radu (engl. *Computer Supported Cooperative Work*), kraće CSCW, predstavlja širi termin koji pokriva sve aplikacije koje olakšavaju suradnju među ljudima. Prema [6] CSCW aplikacije se mogu podijeliti na sinkrone i asinkrone. Sinkrone CSCW aplikacije omogućuju grupni rad koji se provodi u istom trenutku. Asinkrone CSCW aplikacije omogućuju grupni rad (dogovaranje) koje se ne provodi u istom trenutku (npr. e-mail, blogovi, ...). Dogovaranje termina sastanka se ostvaruje asinkronom CSCW aplikacijom tako da se ljudi, koji trebaju doći na sastanak, izjasne kada im najbolje odgovara održavanje sastanka (osoba se može izjasniti o ponuđenim terminima u bilo kojem trenutku, prije nego se doneše odluka o terminu održavanja sastanka).

U sklopu ovog rada osmišljen je model planiranje sastanaka, te je ostvarena primjenska aplikacija koja omogućuje korisnicima određivanje termina održavanja sastanaka, tako da odabrani termin najbolje pogoduje svim zainteresiranim sudionicima. U nastavku je formaliziran problem odabira najboljeg termina za održavanje sastanka.

### 4.1 Formalizacija problema

Potrebitno je odrediti optimalni termin za održavanje sastanaka.

- Na sastanak treba doći  $n$  osoba:  $o_1, o_2, \dots, o_n$ ,
- Na raspolaganju su  $m$  mogućih termina za održavanje sastanka:  $t_1, t_2, \dots, t_m$ .
- Svih  $n$  osoba je dalo svoje mišljenja o nekim od  $m$  termina (osoba se ne mora izjasniti koliko joj svaki od  $m$  termina odgovara, nego se može izjasniti za samo neke termine). Ako se je osoba  $o_i$  izjasnila koliko joj odgovara termin  $t_j$ , tada je

$b_{ij} = 1$ , inače  $b_{ij} = 0$ . Dakle,  $b_{ij} \in \{0,1\}$  bilježi je li se osoba  $o_i$  izjasnila koliko joj odgovara termin  $t_j$ .

- Ako je  $b_{ij} = 1$ , tada  $v_{ij} \in [-1,1]$  govori koliko osobi  $o_i$  odgovara termin  $t_j$  ( $v_{ij}$  je glas osobe). Ako je  $b_{ij} = 0$ , tada je  $v_{ij} = 0$ . Pritom pozitivna vrijednost  $v_{ij}$  označava da osobi  $o_i$  odgovara termin  $t_j$ , a što je  $v_{ij}$  veći, to osobi  $o_i$  više odgovara termin  $t_j$ . Negativna vrijednost  $v_{ij}$  označava da osobi  $o_i$  termin  $t_j$  ne odgovara, a što je  $v_{ij}$  negativniji to osobi  $o_i$  više ne odgovara termin  $t_j$ .
- $w_i$  je jačina (težina) glasa osobe  $o_i$ . Jačina glasa opisuje koliko je bitno mišljenje osobe  $o_i$ . (Ako je  $w_x$  duplo veći od  $w_y$ , tada se daje duplo veći naglasak mišljenju osobe  $o_x$  nego mišljenju osobe  $o_y$  pri odlučivanju o terminu održavanja sastanka.)

Želimo li odabrati onaj termin koji najviše odgovara svim osobama mogli bi prepostaviti da je logično odabrati termin prema (4-1).

$$\operatorname{argmax}_j \left\{ \sum_{i=1}^n v_{ij} \right\} \quad (4-1)$$

Time biramo onaj termin koji ima najpozitivniji zbroj mišljenja korisnika o tom terminu. Ipak, formula (4-1) je previše jednostavna i ne koristi informacije o jačini glasa pojedine osobe. Uvažimo jačinu glasa osobe  $w_i$ :

$$\operatorname{argmax}_j \left\{ \sum_{i=1}^n v_{ij} w_i \right\} \quad (4-2)$$

Formula (4-2) množi mišljenja osobe  $o_i$  s jačinom glasa te osobe kako bi utjecaj pojedine osobe na zbroj bio proporcionalan jačini glasa te osobe.

Sljedeći problem koji treba razmotriti je razlika zbroja glasova pojedinih osoba (količina mišljenja koje je osoba dala o terminima). Ako su tri osobe (s jednakim jačinama glasa) dale mišljenje o tri termina kao što je prikazano tablicom 4.1, tada najveći zbroj dobiva termin  $t_1$ , međutim, logično bi bilo da se za održavanje sastanka odabere termin  $t_2$  jer najbolje odgovara i osobi  $o_2$  i osobi  $o_3$ .

**Tab. 4.1.** Primjer podataka dobivenih od korisnika

	$o_1$	$o_2$	$o_3$	$\sum_{i=1}^3 v_{ij}$
$t_1$	1			1
$t_2$	-1	1	0.5	0.5
$t_3$	-1			0

Zbroj iz (4-2) za termin  $t_1$  poprima najveću vrijednost zato što se osoba  $o_1$  izjasnila o svim terminima; te zato znatno više utječe na donošenje odluke nego utječu ostale osobe, koje su se izjasnile samo o jednom terminu. Utjecaj svih osoba (s istom jačinom glasa) pri donošenju odluke o terminu održavanja sastanka treba biti jednak. To možemo postići tako da skaliramo zbroj apsolutnih vrijednosti svih glasova pojedine osobe na istu vrijednost.

Uvodimo skaliran glas  $\tilde{v}_{ij}$  pojedine osobe  $o_i$  o terminu  $t_j$ , s vrijednošću definiranom prema (4-3).

$$\tilde{v}_{ij} = \frac{v_{ij}}{\sum_{j=1}^m |v_{ij}|} \quad (4-3)$$

Time za svaku osobu  $o_i$  vrijedi:

$$\sum_{j=1}^m |\tilde{v}_{ij}| = 1 \quad (4-4)$$

Odnosno, zbroj svih skaliranih glasova osobe ima istu vrijednost.

Tablica 4.2 prikazuje izbor najpovoljnijeg termina održavanja sastanka ako se koristi formula (4-5).

$$\operatorname{argmax}_j \left\{ \sum_{i=1}^n \tilde{v}_{ij} w_i \right\} \quad (4-5)$$

**Tab. 4.2.** Primjer odabira termina pomoću skaliranih podataka

	$o_1$	$o_2$	$o_3$	$\sum_{i=1}^3 \tilde{v}_{ij}$
$t_1$	0.33			0.33
$t_2$	-0.33	1	1	1.67
$t_3$	-0.33			-0.33

Iz tablice 4.2 se vidi da termin  $t_2$  ima najveći zbroj te je zato najbolji termin za održavanje sastanka, dok termin  $t_3$  ima najmanji zbroj te je najnepovoljniji (što je također logično).

Ipak, pristup sa skaliranjem absolutnog zbroja glasova pojedine osobe ne potiče korisnike sustava da se dovoljno detaljno izjasne o tome koliko im koji termin odgovara – jer nisu na nikakav način nagrađeni za trud koju ulažu pri davanju mišljenja o svim terminima. Zato je potrebno nagraditi korisnike shodno tome za koliko termina su dali glas.

Ako se osoba  $o_i$  izjasnila o nekom terminu  $t_j$ , tada je  $b_{ij} = 1$ , a ako se osoba nije izjasnila onda je  $b_{ij} = 0$ . Ukupan broj termina za koji je osoba dala glas definiramo prema (4-6).

$$B_i = \sum_{j=1}^m b_{ij} \quad (4-6)$$

Uvodimo nagrađenu jačinu težine glasa  $W_i$  koja predstavljanja jačinu glasa  $w_i$  uvećanu ovisno o broju termina za koje je korisnik dao glas.  $W_i$  se definira prema (4-7).

$$W_i = w_i(1 + f(B_i)) \quad (4-7)$$

Pritom je  $f$  funkcija koja odlučuje o veličini nagrade ovisno o  $B_i$ . Za  $f$  je pogodno koristiti (4-8).

$$f = k\sqrt{B_i} \quad (4-8)$$

Pritom je  $k$  konstanta kojom se podešava utjecaj nagrade na jačinu glasa (veći  $k$  rezultira većom nagradom). Konstantu  $k$  je pogodno postaviti na  $k = 0.08$ .

Dakle,  $W_i$  se može definirati prema (4-9).

$$W_i = w_i(1 + k\sqrt{B_i}) = w_i + w_i k \sqrt{B_i} \quad (4-9)$$

Sada se odabir optimalnog termina vrši prema (4-10).

$$\operatorname{argmax}_j \left\{ \sum_{i=1}^n \tilde{v}_{ij} W_i \right\} \quad (4-10)$$

Za kraj razmotrimo tablicu 4.3 u kojoj je pet osoba dalo glas o dva termina – glasovi nisu skalirani, ali je zbroj absolutne vrijednosti glasova svake osobe ista, te sve osobe imaju istu jačinu glasa  $w_i$ .

**Tab. 4.3.** Primjer podataka

	$o_1$	$o_2$	$o_3$	$o_4$	$o_5$	$\sum_{i=1}^5 v_{ij}$	$\sum_{i=1}^5 \tilde{v}_{ij}$
$t_1$	1	1	1	1	-1	3	1.875
$t_2$	0.6	0.6	0.6	0.6	0.6	3	1.875

Iz tablice se vidi da oba termina imaju jednak velik zbroj, te bi se moglo prepostaviti da su jednako dobri za održavanje sastanka. Ipak, termin  $t_1$  osobi  $o_5$  nimalo ne odgovara, dok termin  $t_2$  svima poprilično odgovara.

Logično je odabrati onaj termin koji svima odgovara umjesto onog koji nekom nimalo ne odgovara. To se može postići ako radije biramo one termine o kojima su sve osobe dale međusobno sličan glas.

Raspršenost glasova za pojedini termin možemo statistički mjeriti standardnom devijacijom (4-11).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i - \bar{\tilde{v}}_{ij} \bar{W}_i)^2} \quad (4-11)$$

Pri čemu je  $\bar{\tilde{v}}_{ij} \bar{W}_i$  prosječan glas za termin  $t_j$ , definiran izrazom (4-12).

$$\overline{\tilde{v}_{ij}W_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i \quad (4-12)$$

Budući da preferiramo termine s manjom raspršenosti glasova (time i manjom standardnom devijacijom glasova za taj termin), možemo odabir termina vršiti izrazom (4-13).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i}{n} - q \sigma_j \right\} \quad (4-13)$$

Pri čemu zbroj iz formule (4-10) dijelimo s brojem osoba  $n$  – kako bi nam razlomak iz formule (4-13) dao prosječan glas za pojedini termin  $t_j$ .

Konstanta  $q$  određuje koliko nam je pri odabiru termina bitno da su glasovi međusobno slični (veći  $q$  rezultira većim preferiranjem termina s manjom standardnom devijacijom glasova). Izraz (4-13) prosječni glas za pojedini termin umanjuje za  $q$  pomnožen sa standardnom devijacijom glasova za taj termin. Praktično je za vrijednost konstante  $q$  uzeti vrijednost iz intervala  $[0.25, 0.35]$ .

Izraz (4-13) predstavlja konačan način na koji se odabire najpovoljniji termin za održavanje sastanka. Taj termin je određen izrazom (4-14).

$$t = \operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i}{n} - q \sigma_j \right\} \quad (4-14)$$

#### 4.1.1 Dodatni parametri modela

U nastavku se u model uvode dodatni parametri koji omogućuju bolji odabir termina.

Smisleno je omogućiti pojedinoj osobi da se izjasni koliko joj je sastanak bitan – kako bi se pri odabiru najpovoljnijeg termina za održavanje sastanka odabrao onaj termin koji što više pogoduje osobama koje su zainteresirane za sastanak (sastanak im je bitan). Posljedica takvog odabira je očekivano kvalitetniji sastanak jer je termin prilagođen osobama koje žele doći na sastanak. Osoba  $o_i$  zainteresiranost za sastanak izražava parametrom  $z_i$  iz intervala  $[0,1]$ .

Pri organiziranju sastanka ponekad je poželjno sastanak održati u terminu koji što bolje odgovara osobama dobro upoznatim sa tematikom koja će se na sastanku obraditi (kako

bi sastanak bio što učinkovitiji). Osoba  $o_i$  može izraziti poznavanje tematike parametrom  $p_i$  iz intervala  $[0,1]$ .

Dodatac parametar koji pokazuje motiviranost osobe za sastanak je trud koji osoba ulaže pri generiranju svog mišljenja o pojedinom terminu. Moguće je pratiti trajanje popunjavanja forme, broj stisnutih tipki, zbroj dužina svih poteza mišem, i slične aktivnosti prilikom popunjavanja forme s terminima za održavanje sastanka. Logično je prepostaviti da je osobi koja uloži puno truda pri procjeni termina koji joj odgovaraju, stalo do toga da dođe na sastanak, te zato treba njeni mišljenje uvažiti više nego mišljenje neke druge osobe koja je bez previše razmišljanja brzo popunila formu. Trud osobe  $o_i$  pri popunjavanju forme opisuje pozitivan broj  $a_i$ .

Dakle, uvodimo tri nova parametra koji dodatno opisuju osobu  $o_i$ .

- Parametar  $z_i \in [0,1]$  opisuje zainteresiranost osobe  $o_i$  za sastanak.
- Parametar  $p_i \in [0,1]$  opisuje koliko osoba  $o_i$  poznaje tematiku sastanka.
- Parametar  $a_i \in [0, +\infty)$  opisuje trud koji je osoba  $o_i$  uložila pri generiranju svog mišljenja o pojedinom terminu.

Izraz za odabir termina koji koristi parametar  $z_i$  prikazan je izrazom (4-15).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i z_i}{n} - q \sigma_j \right\} \quad (4-15)$$

U izrazu (4-15)  $z_i$  uzrokuje smanjenje zbroja, pa je zato potrebno prilagoditi i izraz za računanje standardne devijacije  $\sigma_j$  kao što je prikazano izrazom (4-16), pri čemu je  $\overline{\tilde{v}_{ij} W_i z_i}$  definiran izrazom (4-17).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i z_i - \overline{\tilde{v}_{ij} W_i z_i})^2} \quad (4-16)$$

$$\overline{\tilde{v}_{ij} W_i z_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i z_i \quad (4-17)$$

Ipak, uvrštavanje  $z_i$  direktno u (4-15) uzrokovat će da se osobama sa  $z_i = 0$  potpuno zanemare sve njihove odluke. Zato je poželjno moći pomoću dodatne konstante odrediti koliko  $z_i$  može utjecati na glasove osobe  $o_i$ .

Definiramo  $Z_i$  pomoću (4-18).

$$Z_i = uz_i + 1 - u \quad (4-18)$$

U izrazu (4-18) konstanta  $u \in [0,1]$  odlučuje koliko parametar  $z_i$  utječe na glasove osoba ako za odlučivanje koristimo izraz (4-19).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i}{n} - q \sigma_j \right\} \quad (4-19)$$

U izrazu (4-18) konstanta  $u$  odlučuje na koliko mali interval će se skalirati početni interval  $[0,1]$  parametra  $z_i$ . Npr. ako je  $u = 0.2$  tada  $Z_i \in [0.8,1]$ .

U izrazu (4-19) treba računati  $\sigma_j$  prilagodbom izraza (4-16) i (4-17) tako da se umjesto  $z_i$  koristimo  $Z_i$ .

Uvedimo sada parametar  $p_i$ . Izraz za odabir termina koji koristi parametar  $p_i$  ( $p_i$  opisuje poznavanje tematike) prikazan je izrazom (4-20).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i p_i}{n} - q \sigma_j \right\} \quad (4-20)$$

U izrazu (4-20) potrebno je standardnu devijaciju  $\sigma_i$  računati pomoću izraza (4-21), budući da množenje s parametrom  $p_i$  uzrokuje smanjenje zbroja, te je potrebno prilagoditi i izraz za računanje standardne devijacije uvažujući smanjene jačine glasova koji nastaju množenjem s parametrom  $p_i$ . U izrazu (4-21) se  $\overline{\tilde{v}_{ij} W_i Z_i p_i}$  definira izrazom (4-22).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i Z_i p_i - \overline{\tilde{v}_{ij} W_i Z_i p_i})^2} \quad (4-21)$$

$$\overline{\tilde{v}_{ij} W_i Z_i p_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i Z_i p_i \quad (4-22)$$

Uvrštavanje  $p_i$  u izraz (4-20) rezultira poništavanjem glasa osoba koje ne poznaju tematiku (slučaj kada  $p_i = 0$ ). Poželjno je pomoću dodatne konstante  $r$  podesiti utjecaj  $p_i$  na glasove osobe  $o_i$ .

Zato definiramo  $P_i$  pomoću (4-23), slično kao (4-18).

$$P_i = r p_i + 1 - r \quad (4-23)$$

U izrazu (4-23) konstanta  $r \in [0,1]$  odlučuje koliko parametar  $P_i$  utječe na glasove osoba ako za odlučivanje koristimo izraz (4-24), koji umjesto  $p_i$  koristi  $P_i$ .

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i P_i}{n} - q \sigma_j \right\} \quad (4-24)$$

U izrazu (4-23) konstanta  $r$  odlučuje na koliko mali interval treba skalirati početni interval  $[0,1]$  parametra  $p_i$ , kako bi se smanjio utjecaj parametra  $p_i$  na glas osobe  $o_i$ . Npr. ako je  $r = 0.1$  tada  $P_i \in [0.9,1]$ .

U izrazu (4-24) treba računati  $\sigma_j$  prilagodbom izraza (4-21) i (4-22) tako da se umjesto  $p_i$  koristimo  $P_i$ .

Uvedimo još parametar  $a_i$  koji opisuje trud koji je osoba uložila pri generiranju svojih glasova o terminima. Ugradnjom parametra  $a_i$  u izraz za odabir termina dobivamo izraz (4-25).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i P_i a_i}{n} - q \sigma_j \right\} \quad (4-25)$$

U izrazu (4-25) potrebno je standardnu devijaciju  $\sigma_i$  računati pomoću izraza (4-26), pri čemu se  $\overline{\tilde{v}_{ij} W_i Z_i P_i a_i}$  definira izrazom (4-27).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i Z_i P_i a_i - \overline{\tilde{v}_{ij} W_i Z_i P_i a_i})^2} \quad (4-26)$$

$$\overline{\tilde{v}_{ij} W_i Z_i P_i a_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i Z_i P_i a_i \quad (4-27)$$

Uvrštavanje  $a_i$  u izraz (4-25) rezultira množenjem glasa s trudom osobe, koji je iz intervala  $a_i \in [0, +\infty)$ . Želimo li moći upravljati utjecajem parametra  $a_i$  na glasove osoba, potrebno je interval  $[0, +\infty)$  preslikati na interval  $[0,1]$ . To činimo definiranjem  $A_i$  izrazom (4-28).

$$A_i = 1 - \frac{1}{s a_i} \quad (4-28)$$

U izrazu (4-28) vrijedi  $A_i \in [0,1]$  ako  $s a_i > 1$ . Pretpostavimo da je  $a_i \gg 1$ . Utjecaj  $A_i$  na odabir termina, definiranog izrazom (4-29), podešavamo parametrom  $s \in [0,1]$ , točnije  $s \in [\frac{1}{\min_i \{a_i\}}, 1]$ . U izrazu (4-28) konstanta  $s$  odlučuje o utjecaju parametra  $a_i$  na glas osobe. Npr. ako je  $s = 0.1$ , tada za  $a_i = 20$  dobivamo  $A_i = 0.5$ , za  $a_i = 100$  dobivamo  $A_i = 0.9$ , dok za  $a_i = 1000$  dobivamo  $A_i = 0.99$ .

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i P_i A_i}{n} - q \sigma_j \right\} \quad (4-29)$$

U izrazu (4-29) potrebno je  $\sigma_j$  računati izrazom (4-30), a  $\overline{\tilde{v}_{ij} W_i Z_i P_i A_i}$  iz (4-30) izrazom (4-31).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i Z_i P_i A_i - \overline{\tilde{v}_{ij} W_i Z_i P_i A_i})^2} \quad (4-30)$$

$$\overline{\tilde{v}_{ij} W_i Z_i P_i A_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i Z_i P_i A_i \quad (4-31)$$

Parametri koji opisuju korisnika:  $z_i \in [0,1]$  (opisuje zainteresiranost osobe  $o_i$  za sastanak),  $p_i \in [0,1]$  (opisuje koliko osoba  $o_i$  poznaje tematiku sastanka) i  $a_i \in [0, +\infty)$  (opisuje trud osoba  $o_i$ ) podložni su manipulacijama:

- Ako osoba zna način na koji se koriste parametri  $z_i$ ,  $p_i$  i  $a_i$  – može namjerno unijeti netočne vrijednosti kako bi se njen glas više uvažio prilikom odabira najboljeg termina za održavanje sastanka.
- Problem uzrokuju i različite karakterne osobine osobe – neka osoba može često izjavljivati da je zainteresirana za sastanak (unoseći velik  $z_i$ ) jer ju mnogo toga zanima, dok će neka druga osoba (koja je po prirodi manje zainteresirana) često unositi male vrijednosti parametra  $z_i$ . Neke osobe su urednije od drugih i daju više truda detaljima te će takve osobe vjerojatnije imati veću vrijednost parametra  $a_i$ .
- Na parametar  $a_i$  utječe i općenito iskustvo korištenje računala; nedovoljno poznavanje prirodnog jezika koji sustav koristi za komunikaciju s korisnikom; vrsta uređaja koja se koristi za pristup aplikaciji (mobilni uređaji ili stolno računalo) i sl.

Svi navedeni problemi uzrokuju trajni pomak prosjeka unesenih parametara (uneseni parametri pojedine osobe za različite sastanke neće činiti pravilnu Gaussovou razdiobu).

Rješenje je uvođenje novih parametara koji definiraju prosječnu vrijednost parametara koju pojedina osoba unosi u sustav (korisnik sustava se prati duže vrijeme kako bi se znala prosječna vrijednost parametara koje unosi). Zato uvodimo tri nova parametra:

- $\bar{Z}_l$  je prosječna zainteresiranost osobe  $o_i$
- $\bar{P}_l$  je prosječno poznavanje tematike osobe  $o_i$
- $\bar{A}_l$  je prosječan trud pri generiranju glasova osobe  $o_i$

Prosjek bi mogli jednostavno definirati pomoću izraza (4-32), pritom je  $x$  neki parametar za koji računamo prosjek.

$$\bar{x} = \frac{1}{l} \sum_{i=1}^l x_l \quad (4-32)$$

U izrazu (4-32) računa se prosjek za  $l$  unosa parametra  $x$ , za koje znamo da pripadaju istoj osobi. Taj izraz počinje biti smislen tek kada  $l$  poprima veće vrijednosti. Ako imamo samo jedan unosa nekog parametra  $x$  s vrijednošću  $x_1 = 0.9$  nema smisla proglašiti taj broj prosjekom, budući da još nemamo dovoljno velik uzorak (imamo samo jedan  $x$ ). Ipak, budući da znamo interval unutar kojega se  $x$  mora nalaziti

( $x \in [0,1]$ , ako se radi i parametru  $Z_i$ ,  $P_i$  ili  $A_i$ ), možemo očekivati prosječnu vrijednost 0.5. Zato je smisleno koristiti Laplaceovo zaglađivanje (engl. *Laplace Smoothing*) [11] kako bi prvih nekoliko vrijednosti parametra  $x$  bili bliže očekivanoj vrijednosti. Laplaceovo zaglađenje  $h$ -tog reda uz očekivanje  $\epsilon$  definira se izrazom (4-33).

$$LS(h): \bar{x} = \frac{\sum_{i=1}^l x_i + h\epsilon}{l + h} \quad (4-33)$$

Uz očekivanje  $\epsilon = 0.5$  i odabir Laplaceovog zaglađenja drugog reda  $\bar{Z}_l$ ,  $\bar{P}_l$  i  $\bar{A}_l$  se izračunavaju prema izrazima (4-34), (4-35) i (4-36).

$$\bar{Z}_l = \frac{\sum_{j=1}^l z_{ij} + 1}{l + 2} \quad (4-34)$$

$$\bar{P}_l = \frac{\sum_{j=1}^l p_{ij} + 1}{l + 2} \quad (4-35)$$

$$\bar{A}_l = \frac{\sum_{j=1}^l a_{ij} + 1}{l + 2} \quad (4-36)$$

Pritom  $z_{ij}$ ,  $p_{ij}$  i  $a_{ij}$  predstavljaju  $j$ -ti unos parametra osobe  $o_i$ .

Poznavajući prosječnu vrijednost parametra možemo razmotriti parametar koji je osoba dala u odnosu na vrijednost parametra koju obično daje. Npr. ako osoba  $o_1$  unese  $Z_1 = 0.4$ , a na temelju prijašnjih unosa te osobe znamo da je  $\bar{Z}_1 = 0.3$ , te osoba  $o_2$  unese  $Z_2 = 0.6$ , a ne temelju prijašnjih unosa osobe  $o_2$  znamo da je  $\bar{Z}_2 = 0.8$  – tada možemo zaključiti da je osoba  $o_1$  zainteresiranija za sastanak nego obično, te da je osoba  $o_2$  manje zainteresirana nego obično, te zato treba više uvažiti mišljenje osobe  $o_1$ .

Zato definiramo usrednjene vrijednosti  $\tilde{Z}_l$ ,  $\tilde{P}_l$  i  $\tilde{A}_l$ , izrazima: (4-37), (4-38) i (4-39).

$$\tilde{Z}_l = \begin{cases} Z_i \frac{0.5}{\bar{Z}_l}, & Z_i \leq \bar{Z}_l \\ (Z_i - \bar{Z}_l) \frac{0.5}{1 - \bar{Z}_l} + 0.5, & Z_i > \bar{Z}_l \end{cases} \quad (4-37)$$

$$\tilde{P}_i = \begin{cases} P_i \frac{0.5}{\bar{P}_i}, & P_i \leq \bar{P}_i \\ (P_i - \bar{P}_i) \frac{0.5}{1 - \bar{P}_i} + 0.5, & P_i > \bar{P}_i \end{cases} \quad (4-38)$$

$$\tilde{A}_i = \begin{cases} A_i \frac{0.5}{\bar{A}_i}, & A_i \leq \bar{A}_i \\ (A_i - \bar{A}_i) \frac{0.5}{1 - \bar{A}_i} + 0.5, & A_i > \bar{A}_i \end{cases} \quad (4-39)$$

Osobe iz prethodnog primjera imaju usrednjenu zainteresiranost:  $\widetilde{Z}_1 = 0.571$  i  $\widetilde{Z}_2 = 0.375$ , što jasno pokazuje da je osoba  $o_1$  zainteresirana za sastanak nego osoba  $o_2$ .

Konačan izraz za odabir optimalnog termina za organiziranje sastanka daje (4-40).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i}{n} - q \sigma_j \right\} \quad (4-40)$$

U izrazu (4-40) potrebno je  $\sigma_j$  računati izrazom (4-41), a  $\overline{\tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i}$  iz (4-41) izrazom (4-42).

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i - \overline{\tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i})^2} \quad (4-41)$$

$$\overline{\tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i} = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i \tilde{Z}_i P_i \tilde{A}_i \quad (4-42)$$

U izrazu (4-40) ne koristi se usrednjena vrijednost poznavanja tematike ( $\widetilde{P}_i$ ) jer nema smisla usrednjavati poznavanje tematike, budući da uzastopan unos velikog iznosa parametra  $p_i$  govori o tome da se radi o iskusnom korisniku.

Konačan izraz (4-40) ovisi o sljedećim ulaznim parametrima:

- $n$  – broj osoba,
- $m$  – broj termina,
- $b_{ij}$  – je li osoba  $o_i$  dala glas za termin  $t_j$ ,
- $v_{ij}$  – iznos glasa osobe  $o_i$  za termin  $t_j$ ,

- $w_i$  – jačina glasa osobe  $o_i$ ,
- $z_i$  – zainteresiranost osobe  $o_i$  za sastanak.
- $p_i$  – koliko osoba  $o_i$  poznaje tematiku sastanka.
- $a_i$  – trud osobe  $o_i$  pri unošenju mišljenja
- $\bar{Z}_i$  – prosječna zainteresiranost osobe  $o_i$
- $\bar{A}_i$  – prosječan trud pri generiranju glasova osobe  $o_i$

I konstantama  $k, q, u, r$  i  $s$ .

### 4.1.2 Daljnje razmatranje

Treba primijetiti da poznavanje tematike ponekad nije koristan parametar (npr. pri organiziranju termina opuštenog druženja). Štoviše, ponekad je cilj da termin sastanaka što bolje odgovara osobama koje nisu upoznate s tematikom kojom se sastanak bavi (npr. tečaj plesa je korisniji osobama koje ne znaju plesati nego osobama koje znaju plesati).

Postoje mnogi uzroci šuma koji rezultiraju netočnim parametrima koje daje korisnik. Neki od njih su:

- Koliko se osobi žuri – više vremena na raspolaaganju rezultira dužim korištenjem i točnijim informacijama.
- Doba dana – neke osobe su odmornije ujutro dok su druge odmornije navečer.
- Osobni uzroci – netko će biti umoran prije nego popije kavu, dok će nakon toga biti odmoran i dobro raspoložen.
- Iskustvo korištenja sustava – osobe koje duže koriste sustav postanu efikasnije u korištenju sustava, te im treba manje truda da postignu istu radnju. Osobe se također razlikuju i po brzini učenja korištenja sustava.

Zato treba pažljivo odabrati konstante  $u, r$  i  $s$  koje definiraju utjecaj korisničkih parametara na jačinu njihovog glasa.

Također valja primijetiti da su neki parametri u korelaciji, npr. trud ovisi o tome koliko je termina osoba popunila.

Neki od problema mogu se riješiti uvođenjem dodatnih parametara:

- Doba dana

- Dob osobe, zanimanje i sl. osobni podaci
- Koliko dugo se koristi sustav:
  - ukupno vrijeme korištenja sustava
  - broj popunjениh rasporeda
  - vrijeme proteklo od prvog korištenja sustava
  - vrijeme proteklo od posljednjeg korištenja sustava
- Praćenje prosječnih vrijednosti parametara u ovisnosti o svim drugim parametrima.
- Korisnički trud uložen pri korištenju pojedine komponente grafičkog sučelja sustava – neka osoba će se par puta predomisliti u vezi nekog termina, što čini njeno mišljenje o tom specifičnom terminu manje bitnim).
- Vjerovatnost pokušaja zloupotrebe sustava – ako se prosječna brzina korištenja sustava povećava te u nekom trenutku počne trajno opadati, moguće je da korisnik pokušava zloupotražiti sustav kako bi se njegov glas smatrao bitnijim (ili je korisnik promijenio način na koji koristi sustav, npr. koristi ga s novog mobilnog uređaja).

# 5. Primjenski program

## 5.1 Opis primjenskog programa

Na Google App Engine oblaku računala implementiran je primjenski program u Java radnom okružju. Primjenski program je stvoren pomoću NetBeans 7.1 integriranog razvojnog okružja s podrškom za Java Enterprise Edition (Java EE) tehnologije, te uz pomoć Google App Engine SDK 1.7.1.

Primjenski program implementira model planiranja sastanaka opisan izrazom (5-1).

$$\operatorname{argmax}_j \left\{ \frac{\sum_{i=1}^n \tilde{v}_{ij} W_i Z_i}{n} - q \sigma_j \right\} \quad (5-1)$$

Pri čemu je  $Z_i$  definiran izrazom (5-2), a  $\sigma_j$  je definiran izrazima (5-3) i (5-4), dok su izrazi za ostale vrijednosti definirane u četvrtom poglavlju.

$$Z_i = u z_i + 1 - u \quad (5-2)$$

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{v}_{ij} W_i Z_i - \bar{\tilde{v}}_{ij} \bar{W}_i \bar{Z}_i)^2} \quad (5-3)$$

$$\bar{\tilde{v}}_{ij} \bar{W}_i \bar{Z}_i = \frac{1}{n} \sum_{i=1}^n \tilde{v}_{ij} W_i Z_i \quad (5-4)$$

Neizvedeni parametri koji se koriste za odlučivanje najboljeg termina za održavanje sastanka su sljedeći:

- $n$  – broj osoba,
- $m$  – broj termina,
- $b_{ij}$  – je li osoba  $o_i$  dala glas za termin  $t_j$ ,
- $v_{ij}$  – iznos glasa osobe  $o_i$  za termin  $t_j$ ,
- $w_i$  – jačina glasa osobe  $o_i$ ,
- $z_i$  – zainteresiranost osobe  $o_i$  za sastanak.

I konstantama  $k$ ,  $q$  i  $u$ .

Vrijednosti navedenih parametara dobivaju se kroz grafičko korisničko sučelje sustava.

Sustav za rad koristi sljedeće tehnologije:

- Java 6 klase, objekti i enumeracije (\*.java)
- Java Servlet i Java Filter (\*.java)
- Java ServerPages, JSTL i Expression Language (\*.jsp)
- HTML 5 (\*.html)
- CSS 3 (\*.css)
- JavaScript i jQuery (\*.js)
- Konfiguracijske datoteke (\*.xml)

Sustav se sastoji od:

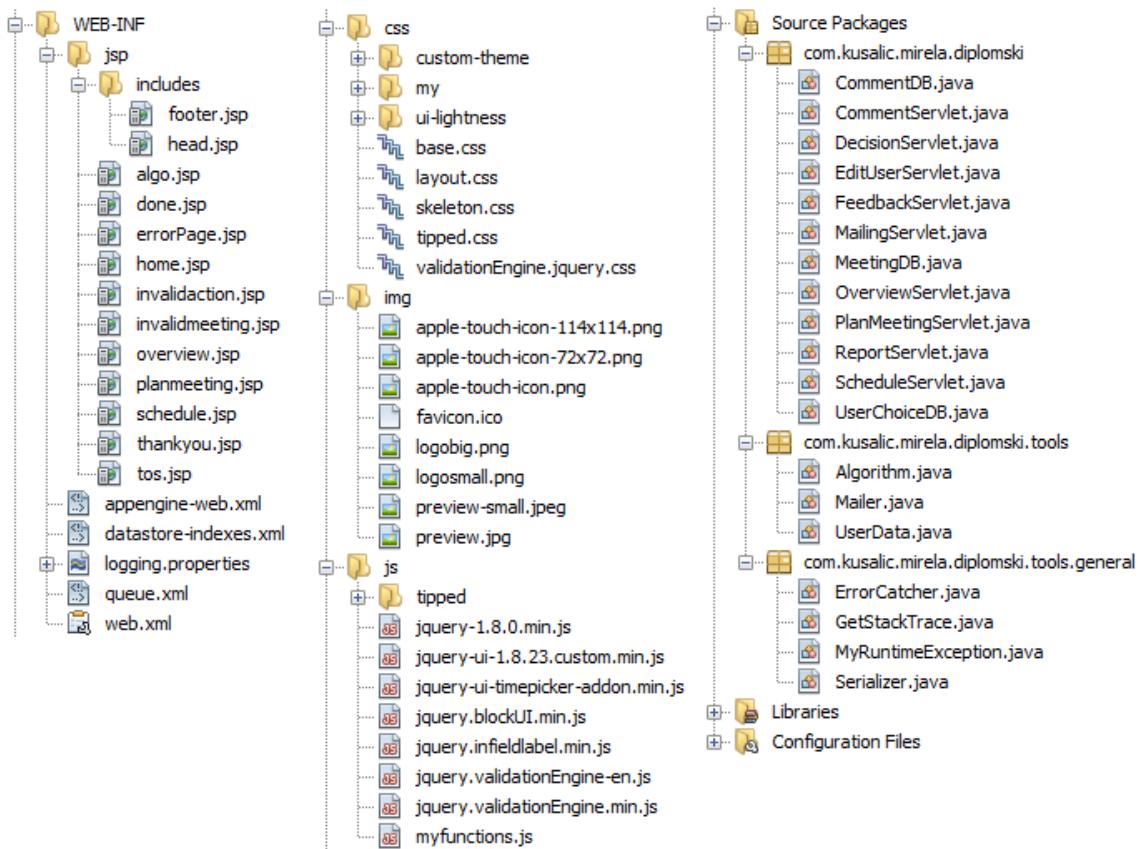
- Četiri konfiguracijske datoteke s ukupno 190 linija koda.
- Pozadinska logika i pohrana podataka čine 19 datoteka s ukupno 2000 linija koda.
- HTML, JSP, JSTL i EL čini ukupno 820 linija koda.
- JavaScript i jQuery čine 810 linija koda.
- CSS se sastoji od 780 linija koda.

Naveden je samo kod koji je napisan isključivo za ostvarenje ovog sustava, sav kod uključen u obliku pomoćnih datoteka i priključaka (engl. *plug-in*) nije ubrojen.

Raspored bitnih datoteka koji čine sustav prikazan je na slici 5.1.

- Lijevo na slici 5.1 prikazane su datoteke koje pružaju dinamički sadržaj klijentu (engl. *front-end*), te konfiguracijske datoteke aplikacije. Dinamički sadržaj zapisan je u \*.jsp datotekama koje sadrže JSP, JSTL i EL programsku sintaksu za implementaciju programske logike, te HTML i JavaScript sintaksu za opis strukture internet stranice i njenih dinamičkih svojstava kod klijenta. \*.jsp datoteke su grupirane u samostojеće stranice (u *jsp* direktorije) i pomoćni sadržaj (u *includes* direktoriju). Unutar WEB-INF direktorija nalaze se i konfiguracijske datoteke u \*.xml i \*.properties formatu. *web.xml* je opisnik aplikacije čiji izgled definira Java Servlet standard, te opisuje (među ostalom) mapiranje prostora aplikacije poServletima koji odgovaraju na korisničke

zahtjeve. *appengine-web.xml* je glavna konfiguracijska datoteka za App Engine aplikaciju. *datastore-index.xml* definira indekse po kojima se pregledavaju trajno zapisani podaci, dok *queue.xml* definira redove za Task Queue API. *logging.properties* definira pravila logiranja informacija i grešaka.



**Sl. 5.1. Prikaz datoteka sustava**

- U sredini slike 5.1 prikazane su datoteke koje pružaju statički sadržaj klijentu. Datoteke su grupirane u direktorije prema tipu. U direktoriju *js* nalaze se JavaScript datoteke. Datoteke s nazivom *jquery\** definiraju jQuery jezgru ili jQuery priključke (npr. *jquery-ui-timepicker-addon.min.js* je komprimirana verzija jQuery priključka koji definira iskočni kalendar sa odabirom datuma i vremena). *css* direktorij sadrži datoteke sa CSS sintaksom koja opisuje grafički dizajn stranice. Osnovni kostur stranice opisuje *skeleton.css*, *base.css* provodi glavni reset prikaza svih HTML tagova, dok *layout.css* opisuje dizajn stranice za stolne i mobilne uređaje. Ostale datoteke i direktoriji opisuju grafički dizajn jQuery komponenti i priključaka. *img* direktorij sadrži statičke slike korištene na stranici.

- Desno na slici 5.1 prikazane su datoteke koje čine pozadinsku logiku i pohranu podataka (engl. *back-end*). Servleti i klase za pristup pohrani podataka (*\*Servlet.java* i *\*DB.java*) grupirani su u paket *com.kusalic.mirela.diplomski*. Algoritam za izračun najboljeg termina za održavanje sastanka (prema modelu opisanom u četvrtom poglavlju) nalazi se u datoteci *Algoritam.java*. Paketi *com.kusalic.mirela.diplomski.tools* i *com.kusalic.mirela.diplomski.tools.general* sadrže pomoćne mehanizme za slanje elektroničke pošte, hvatanje i zapis grešaka u radu sustava, te za pohranjivanje podataka.

## 5.2 Korištenje programa

Primjenski program je dostupan na internet adresi:

**www.besttimetomeet.com**

(također i na odgovarajućoj \*.appspot.com domeni, te na „goloj“ domeni (engl. *naked domain*) s kojoj se *http* zahtjevi preusmjeruju na *www* poddomenu).

Nakon što korisnik posjeti internet stranicu ostvarenog sustava prikaže se korisničko sučelje sa slike 5.2.

The screenshot shows the homepage of the Best Time To Meet website. At the top left is a logo featuring a clock face with the text "BEST TIME TO MEET". To the right is a quote: "Plan your **business** or casual meeting. Let the group choose the best time to meet!" Below the quote, there's a section titled "Site snapshots:" showing three screenshots of the website interface. To the right of these snapshots is a bulleted list of features: "Very easy to use", "No registration", "Perfectly chosen term", "Visual representation of user's opinions", and "You can: Give users priorities, Track when users visit the site, Track what users submit, and a lot more.". Below this list are three numbered steps: 1. Create plan, 2. People fill-in, 3. Perfect term is calculated. To the right of these steps is a large blue button with the word "Go!". At the bottom right is a feedback form with the placeholder "Help us improve this site, leave some feedback" and a "Send" button. At the very bottom, there's a small note: "By using this website you agree to the Terms of Service. ©2012 Mirela Kusalić".

**Sl. 5.2. Naslovna stranica**

Korisničko sučelje je vrlo dinamičko, ugodno za koristiti i samoopisivo.

- Na naslovniči su ukratko opisane glavne prednosti sustava.
- Kreiranje sastanka započinje se tipkom *Go!*
- U tri kratka koraka prikazano je kako sustav funkcioniра.
- Za lakše upoznavanje sa sustavom na naslovniči se nalazi slika koja prikazuje primjere korištenja sustava.
- Dinamičke komponente grafičkog sučelja reagiraju na akcije mišem, te se pojavljuju odgovarajuća pojašnjenja.
- S naslovnicice se mogu administratoru slati komentari o sustavu.
- Također se može otvoriti stranica s pojašnjenjem funkcioniranja algoritma korištenog za odabir najpogodnijeg termina.

Nakon što korisnik odabere tipku *Go!* prikaže se stranica za kreiranje sastanka, čiji je izgled prikazan na slici 5.3.

Na stranici za kreiranje sastanaka:

- Mogu se navesti mogući termini za održavanje sastanka. Vrijeme i datum termina se odabiru grafičkom komponentom koja prikazuje kalendar za odabir datuma i dva kliznika (engl. *slider*) za odabir vremena. Uz svaki termin mogu se dodati komentari s informacijama vezanim za taj termin (npr. mjesto održavanje sastanka). Može se dodati proizvoljan broj termina, te se višak termina može ukloniti. Raspored unošenja termina nije bitan budući da sustav sortira termina po vremenu prije nego ih prikaže korisniku.
- Navode se informacije o sastanku. Navodi se ime organizatora (koristi se u elektroničkoj pošti koju sustav šalje) i email adresa organizatora (na tu adresu se šalju informacija potrebne za pristup preglednom panelu vezanom za stvoreni sastanak). Ime mora sadržavati barem tri znaka, a email adresa mora biti ispravno oblikovana. Opcionalno se može popuniti i opis sastanka kako bi pozvanici znali o kakvom se sastanku radi. Izračun optimalnog termina za održavanje sastanka može se pokrenuti na tri načina:
  - Kad svi korisnici popune informacije o tome koliko im koji termin odgovara, ili najkasnije u definiran fiksni tren.
  - U fiksnom trenutku.
  - Organizator mora sam pokrenuti odabir optimalnog termina za održavanje sastanka.

Može se odabrati hoće li korisnička odluka biti konačna, ili korisnik može više puta predati svoje mišljenje (u tom slučaju je potrebno odabrati postotak jačine glasa koji se oduzima korisniku kao kazna za svako naknadno predavanje svojeg mišljenja). Organizator može kasnije omogućiti pojedinim osobama ponovno popunjavanje mišljenja bez da su za to kažnjeni smanjivanjem jačine glasa. Dodatno se može zahtijevati slanje podsjetnika da se dođe na sastanak, prije nego se održava sastanak (dostupno samo u *naprednoj* verziji prikaza). Pomoću kliznika (koji se eksponencijalno povećava) odabire se koliko vremena prije sastanka treba poslati podsjetnik.

### Create a meeting

(just fill this page and you're done)



[Basic](#)
[Advanced](#)

Choose possible terms for meeting:

2012-10-01 07:00	monday morning, new building	<a href="#">Delete</a>
2012-10-02 13:00	after lunch, meeting room	<a href="#">Delete</a>
2012-10-05 17:30	late option	<a href="#">Delete</a>
2012-10-04 11:15	<b>Comment (optional)</b>	<a href="#">Delete</a>

[Add more terms](#)

About the meeting:

John Smith	jsmith@email.com
------------	------------------

ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

When to decide which term is the best?

when filled
fixed time
manual

2012-09-30 10:35

Are people allowed to change their opinions after they submit them?

yes
no

penalty for resubmitting is 20%

Remind everyone to go to the meeting, before it starts?

yes
no

3 hours before meeting

Who should come to the meeting?

Name	Email	Add
this person has voice strength: 1		<input type="button" value="Add"/>

List of added people:

- Mark Twain, mtwain@email.com [2.5] [x](#)
- Kingsley Amis, kamis@email.com [0.8] [x](#)
- Jane Austen, jausten@email.com [1] [x](#)
- Isaac Asimov, iasimov@email.com [0.8] [x](#)
- Henry George, hgeorge@email.com [1.5] [x](#)
- Edmund Burke, eburke@email.com [1] [x](#)

(you can add more people later)

[Submit](#)

By using this website you agree to the Terms of Service.  
©2012 Mirela Kusalic

Sl. 5.3. Stranica za planiranje sastanka

- Potrebno je navesti korisnike koji su pozvani na sastanak (oni će elektroničkom poštom dobiti poruku s poveznicom na stranicu gdje mogu popuniti koliko im koji sastanak odgovara). Organizator pri kreiranju sastanka navodi ime i email korisnika. Ime se koristi za identificiranje korisnika u preglednom panelu, te za identificiranje korisnika pri slanju komentara. Email adresa se koristi kako bi se pozvanom korisniku poslao zahtjev da popuni svoja mišljenja u vezi predloženih termina. Organizator može svakoj osobi definirati jačinu glasa (dostupno samo u *naprednoj* verziji prikaza). Jačina glasa je proporcionalna utjecajnosti pojedine osobe na odlučivanje (kako bi sastanak bio prilagođeniji bitnijim osobama). Nakon što se doda pojedina osoba, sučelje omogućuje naknadno (prije stiskanja tipke *Submit*) mijenjanje podataka o toj osobi (i brisanje). Osobe se mogu dodavati i brisati i nakon što se stvori plan sastanka.

Nakon što se popune željeni podaci plan sastanka se kreira stiskanjem tipke *Submit*.

Nakon što se stvori plan sastanka:

- Organizatoru se prikaže pregledni panel.
- Organizatoru se pošalje elektronička pošta koja sadrži poveznicu na pregledni panel.
- Korisnicima se pošalje elektronička pošta koja sadrži poveznicu na stranicu gdje mogu predati svoje mišljenje o pojedinim terminima.

Izgled preglednog panela dan je na slici 5.4.

Pregledni panel:

- Prikazuje informacije o sastanku.
- Služi samo organizatoru i nije ga preporučljivo dijeliti s korisnicima, budući da omogućuje radnje poput odlučivanja o konačnom terminu održavanja sastanka i brisanje cijelog sastanka.
- Panel prikazuje stanje svih korisnika:
  - Jesu li i kada korisnici posjetili stranicu?
  - Jesu li i kada predali svoje mišljenje o sastancima, te koliko puta?

## Overview

(here you can track your invitees)

---

**Hi John Smith!**

Please don't share this page with anyone, because it would give them control over this meeting.  
Please bookmark this page so you don't forget this address (link to this site has been sent to [jsmith@email.com](mailto:jsmith@email.com)).

*Description of the meeting:*

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

The best term for the meeting will be calculated when everyone invited fills in their opinions, or at latest at **2012-09-30 10:35**.  
Everyone will be reminded to go to the meeting **3 hours** before the meeting starts.  
Each time invitee's opinion is resubmitted, his influence on decision is reduced for 20%.

Refresh!

**Opinions about the terms: (from 4 invitees)**

Term	Fitness
2012-10-01 07:00	0.6
2012-10-01 07:00	0.4
2012-10-01 07:00	-0.1
2012-10-01 07:00	-0.3
2012-10-02 13:00	1.0
2012-10-02 13:00	0.7
2012-10-02 13:00	0.2
2012-10-02 13:00	-0.2
2012-10-04 11:15	0.4
2012-10-04 11:15	0.2
2012-10-04 11:15	0.1
2012-10-04 11:15	-0.1
2012-10-05 17:30	0.5
2012-10-05 17:30	-0.1
2012-10-05 17:30	-0.8

<b>Mark Twain</b> , <a href="mailto:mtwain@email.com">mtwain@email.com</a> [2.5]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Kingsley Amis</b> , <a href="mailto:kamis@email.com">kamis@email.com</a> [0.8]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Jane Austen</b> , <a href="mailto:jausten@email.com">jausten@email.com</a> [1.0]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Isaac Asimov</b> , <a href="mailto:iasimov@email.com">iasimov@email.com</a> [0.8]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Henry George</b> , <a href="mailto:hgeorge@email.com">hgeorge@email.com</a> [1.5]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Edmund Burke</b> , <a href="mailto:eburke@email.com">eburke@email.com</a> [1.0]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Add person
Delete Meeting

*Write some comments:*

Send

**Mark Twain**  
(on 2012-09-29 19:54)

I agree with Edmund

**Edmund Burke**  
(on 2012-09-29 19:52)

I don't like the late option, we need more time!

By using [this website](#) you agree to the [Terms of Service](#).  
©2012 Mirela Kusalić

#### Sl. 5.4. Pregledni panel

- Panel omogućava i brisanje korisnika, slanje korisniku elektroničke pošte kako bi ga se podsjetilo da popuni svoje mišljenje o terminima, te resetiranje mišljenja

korisnika kako bi mogao ponovno bez kazne predati svoje mišljenje o terminima.

- Dodavanje novog korisnika omogućava se iskočnim obrascem u koji se unesu informacije o korisniku kojeg se dodaje (korisniku se odmah pošalje elektronička pošta sa zamlobom da popuni svoje mišljenje o predloženim terminima).
- Sadrži mnogo dinamičkih komponenti koje se aktiviraju na radnje mišom.
- Kritične radnje zahtijevaju potvrdu u iskočnom prozoru (za brisanje sastanka i brisanje korisnika).
- Omogućuje grafički pregled mišljenja svih korisnika (grafički prikaz se omogući tek nakon što barem dva korisnika predaju svoje mišljenje; kako bi se očuvala anonimnost korisničkih mišljenja).
- Sučelje omogućuje organizatoru slanje komentara i čitanje komentara koje su napisali drugi korisnicu pozvani na isti sastanak.
- Nakon što se odluči konačan termin održavanja sastanka, organizatoru se omogući popunjavanje izvještaja sa sastanka (samo organizator može predati izvještaj, dok svi drugi mogu čitati izvještaj). Izvještaj se može mijenjati i predavati više puta.

Korisnika se elektroničkom poštom poziva da popuni svoje mišljenje o potencijalnim terminima za održavanje sastanka, te mu se prikaže grafičko sučelje prikazano na slici 5.5.

Sučelje omogućuje korisniku:

- Pregled relevantnih informacija o sastanku.
- Prikaz popisa termina o kojima treba predati svoje mišljenje, korištenjem kliznika.
- Izjašnjavanje svoje zainteresiranosti za sastanak.
- Mogućnost pisanja komentara i čitanja komentara koje su napisali drugi korisnici pozvani na isti sastanak.
- Nakon što se odluči o terminu sastanka, korisniku se na istoj stranici prikazuje i grafički zbirni prikaz mišljenja svih osoba pozvanih na sastanak, te prikaz izvještaja koje organizator popunjava.

## Choose preferred time for a meeting

(just fill this page and you're done)



Help John Smith choose the best time for the meeting!

Description of the meeting:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

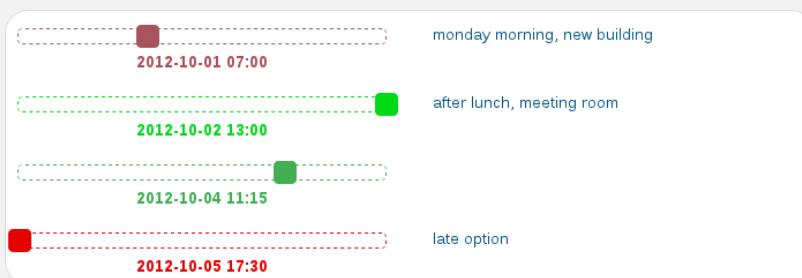
How important is this meeting for you?



important

Please rate each term. Green means it fits you, red means it doesn't:

?



Submit

Write some comments:

Send

Mark Twain  
(on 2012-09-29 19:54)

I agree with Edmund

Edmund Burke  
(on 2012-09-29 19:52)

I don't like the late option, we need more time!

By using this website you agree to the Terms of Service.

©2012 Mirela Kusalić

### Sl. 5.5. Sucelje za korisnika

Nakon što se odluči o terminu održavanja sastanka, svima se pošalje obavijest elektroničkom poštom.

Cijeli sustav je, osim na računalima s velikim ekranom, dostupan i na mobilnim uređajima i na tabletima, te se prilagođava širini ekrana korištenog uređaja. Primjer korištenja sustava s mobilnog uređaja prikazan je na slici 5.6.

**BEST TIME TO MEET**

Plan your **business** or **casual meeting**. Let the group choose the best time to meet!

**Site snapshots:**

- Very easy to use
- No registration
- Perfectly chosen term
- Visual representation of user's opinions
- You can:
  - Give users priorities
  - Track when users visit the site
  - Track what users submit
  - and a lot more.

**1 Create plan**  
**2 People fill-in**  
**3 Perfect term is calculated**

**Go!**

Help us improve this site, leave some feedback

Send

By using this website you agree to the Terms of Service.  
©2012 Mirela Kusalić

### Create a meeting

(just fill this page and you're done)

**Basic** **Advanced**

Choose possible terms for meeting:

2012-10-01 07:00  
monday morning, new building

**Delete**

2012-10-02 13:00  
after lunch, meeting room

**Delete**

2012-09-05 17:30  
late option

**Delete**

2012-09-04 11:15

Comment (optional)

**Delete**

Add more terms

About the meeting:

John Smith  
jsmith@email.com

Opinions about the terms: (from 4 invitees)

When to decide which term is the best?

when filled **fixed time** manual

2012-09-30 10:35

Are people allowed to change their opinions after they submit them?

**yes** **no**

penalty for resubmitting is 20%

Remind everyone to go to the meeting, before it starts?

**yes** **no**

3 hours before meeting

Who should come to the meeting?

Name

Email

Add

this person has voice strength: 1

List of added people:

- Mark Twain, mtwain@email.com [2/5] ×
- Kingsley Amis, kamis@email.co... [0/8] ×
- Jane Austen, jausten@email.co... [1] ×
- Isaac Asimov, iasimov@email.co... [0/8] ×
- Henry George, hgeorge@email.co... [1/5] ×
- Edmund Burke, eburke@email.co... [1] ×

(you can add more people later)

**Submit**

By using this website you agree to the Terms of Service.  
©2012 Mirela Kusalić

### Overview

(here you can track your invitees)

Hi John Smith!

Please don't share this page with anyone, because it would give them control over this meeting. Please bookmark this page so you don't forget this address (link to this site has been sent to jsmith@email.com).

Description of the meeting:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

The best term for the meeting will be calculated when everyone invited fills in their opinions, or at latest at 2012-09-30 10:35. Everyone will be reminded to go to the meeting 3 hours before the meeting starts. Each time invitee's opinion is resubmitted, his influence on decision is reduced for 20%

Refresh!

Opinions about the terms: (from 4 invitees)

When to decide which term is the best?

when filled **fixed time** manual

2012-09-30 10:35

Are people allowed to change their opinions after they submit them?

**yes** **no**

penalty for resubmitting is 20%

Remind everyone to go to the meeting, before it starts?

**yes** **no**

3 hours before meeting

Who should come to the meeting?

Name

Email

Add

this person has voice strength: 1

List of added people:

- Mark Twain, mtwain@email.com [2/5] ×
- Kingsley Amis, kamis@email.co... [0/8] ×
- Jane Austen, jausten@email.co... [1] ×
- Isaac Asimov, iasimov@email.co... [0/8] ×
- Henry George, hgeorge@email.co... [1/5] ×
- Edmund Burke, eburke@email.co... [1] ×

(you can add more people later)

**Submit**

Write some comments:

Send

Mark Twain  
(on 2012-09-29 19:54)

I agree with Edmund

Edmund Burke  
(on 2012-09-29 19:52)

I don't like the late option, we need more time!

By using this website you agree to the Terms of Service.  
©2012 Mirela Kusalić

Sl. 5.6. Mobilne verzije bitnijih sučelja

43

## 5.3 Analiza rada sustava

Sustav je implementiran na Google App Engine oblaku računala zbog sljedećih prednosti:

- App Engine omogućuje skaliranje računalnih resursa potrebnih za posluživanje aplikacije, te prividno omogućuje neograničenu procesnu moć, propusnost (engl. *bandwidth*) i pohranu podataka. Zato je aplikacija spremna, u slučaju naglog rasta popularnosti, posluživati iznimno velik broj korisnika.
- App Engine se koristi kao gotova platformska usluga, te nema potrebe brinuti o održavanju fizičkih servera. Garantirana je dostupnost i sigurnost posluživanja (ista kao drugim Googlom proizvodima), te se ne moramo brinuti o nadogradnji i održavanju korištenog softvera.
- App Engine omogućava korištenje (inače nedostupnih) programskih sučelja koji se izvršavaju na Googlovoj skalabilnoj arhitekturi. Primjenjski program koristi Mail Java API kako bi slao elektroničku poštu pomoću Googleovih servera, koristi Task Queue API kako bi okidao vremenski osjetljive radnje, koristi LogService API kako bi logirao podatke o korištenju sustava i eventualnim greškama, te koristi Googlovo sučelje za pristup trajnoj pohrani podataka).

Sustav uspješno ispunjava zahtjeve:

- Sustav je skalabilan i sposoban paralelno posluživati milijune korisnika.
- Sustav je osnovan na smislenom i matematički utemeljenom modelu za odabir najboljeg termina za održavanje sastanka.
- Sustav je ostvaren u najnovijim internet tehnologijama te ispunjava vizualne i funkcionalne zahtjeve današnjih internet korisnika (na mobilnim i stolnim računalima).
- Sustav je intuitivan i jednostavan za korištenje.

Sustav je primjenjiv za organiziranje sastanaka:

- u sklopu akademske zajednice,
- u poslovnom svijetu,
- te za planiranje svakodnevnih formalnih i neformalnih druženja.

### 5.3.1 Primjer uporabe modela

Primjer 1:

Primjer izračuna algoritma za tri osobe i tri termina:

$$n = 3, \quad m = 3$$

Osobama je organizator dao sljedeće jačine glasova:

$$w = \{1, 1, 2.5\}$$

Osobe su dale mišljenje o terminima:

$$v_1 = \{-1, 0, 1\}$$

$$v_2 = \{1, 1, 1\}$$

$$v_3 = \{0.5, 0.5, 0\}$$

Te zainteresiranost  $z$ :

$$z = \{0, 0.5, 1\}$$

Izračunavamo  $Z$  uz  $u = 0.5$ :

$$Z = \{0.5, 0.75, 1\}$$

Izračunavamo  $\tilde{v}$ :

$$\tilde{v}_1 = \{-0.5, 0, 0.5\}$$

$$\tilde{v}_2 = \{0.333, 0.333, 0.333\}$$

$$\tilde{v}_3 = \{0.5, 0.5, 0\}$$

Za  $B$  vrijedi:

$$B = \{2, 3, 2\}$$

Izračunavamo  $W$  uz  $k = 0.08$ :

$$W = \{1.1131, 1.1386, 2.7828\}$$

Računamo  $\sigma$ :

$$\sigma = \{0.6936, 0.6002, 0.1327\}$$

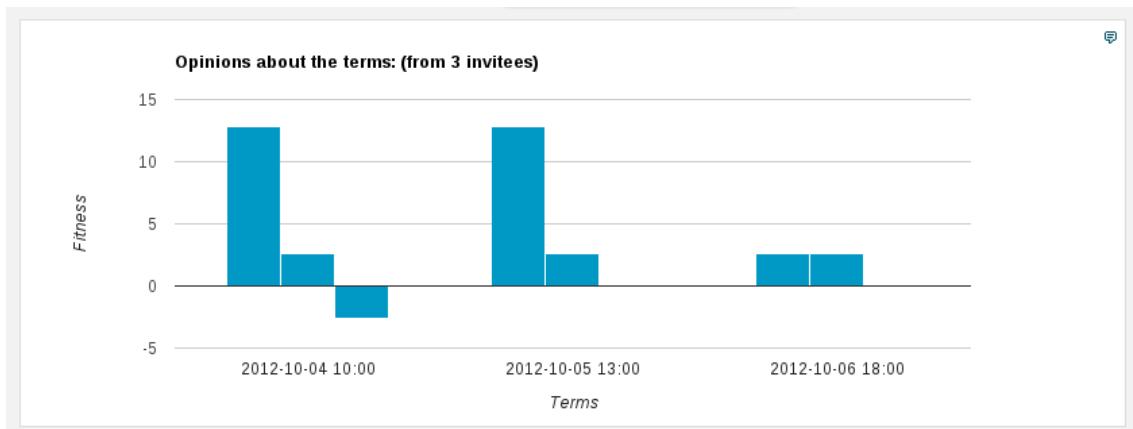
Uz  $q = 0.3$  dobivamo za termine vrijednosti:

$$rezultat = \{0.2578, 0.3786, 0.1478\}$$

Te odabiremo drugi termin budući da ima najveću vrijednost.

Odluka je vrlo logična, te rezultat dobro odgovara informacijama dobivenim od korisnika (i njihovim jačinama glas).

Uz navedene parametre, sustav će prikazati graf sa slike 5.7.



**Sl. 5.7.** Grafički prikaz korisničkih mišljenja za primjer 1  
(mišljenja su skalirana i uvažena je jačina glasa)

### Primjer 2:

Primjer izračuna algoritma za četiri osobe i dva termina:

$$n = 4, \quad m = 2$$

Osobama je organizator dao sljedeće jačine glasova:

$$w = \{1, 1.5, 1, 2\}$$

Osobe su dale mišljenje o terminima:

$$v_1 = \{1, -1\}$$

$$v_2 = \{1, 0\}$$

$$v_3 = \{-1, 0.35\}$$

$$v_4 = \{0.35, 1\}$$

Te zainteresiranost  $z$ :

$$z = \{1, 1, 0, 0.76\}$$

Izračunavamo  $Z$  uz  $u = 0.5$ :

$$Z = \{1, 1, 0.5, 0.879\}$$

Izračunavamo  $\tilde{v}$ :

$$\tilde{v}_1 = \{0.5, -0.5\}$$

$$\tilde{v}_2 = \{1, 0\}$$

$$\tilde{v}_3 = \{-0.741, 0.259\}$$

$$\tilde{v}_4 = \{0.259, 0.741\}$$

Za  $B$  vrijedi:

$$B = \{2, 1, 2, 2\}$$

Izračunavamo  $W$  uz  $k = 0.08$ :

$$W = \{1.1131, 1.62, 1.1131, 2.2263\}$$

Računamo  $\sigma$ :

$$\sigma = \{0.7196, 0.7359\}$$

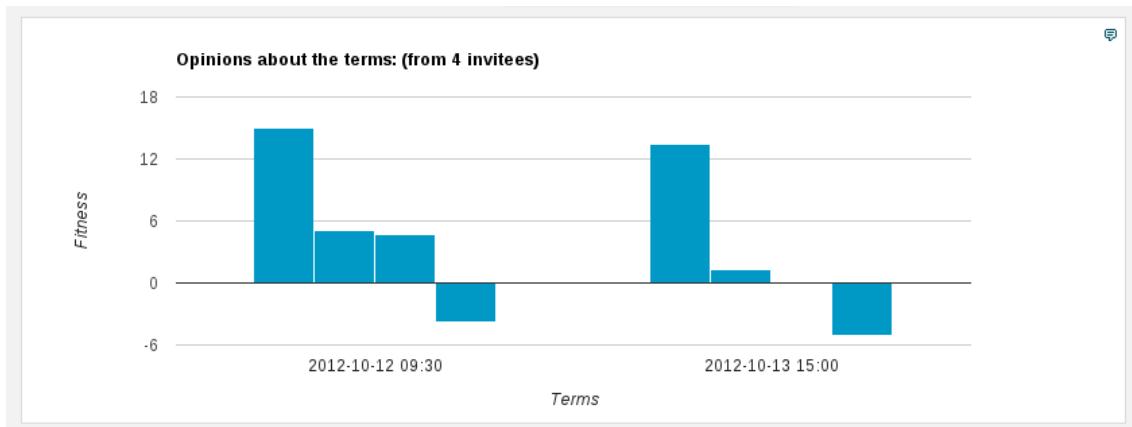
Uz  $q = 0.3$  dobivamo za termine vrijednosti:

$$rezultat = \{0.3525, 0.0389\}$$

Te odabiremo prvi termin budući da ima najveću vrijednost.

Odluka je vrlo logična, drugi termin ima vrijednost sličnu nuli budući da su mišljenja za taj termin vrlo raspršena (velika standardna devijacija).

Uz navedene parametre, sustav će prikazati graf sa slike 5.8.



**Sl. 5.8.** Grafički prikaz korisničkih mišljenja za primjer 2

Primjer 3:

Primjer izračuna algoritma za dvije osobe i četiri termina:

$$n = 2, \quad m = 4$$

Osobama je organizator dao sljedeće jačine glasova:

$$w = \{1, 1\}$$

Osobe su dale mišljenje o terminima:

$$v_1 = \{1, 1, -0.4, -1\}$$

$$v_2 = \{-1, 1, -0.45, 1\}$$

Te zainteresiranost  $z$ :

$$z = \{0.5, 1\}$$

Izračunavamo  $Z$  uz  $u = 0.5$ :

$$Z = \{0.75, 1\}$$

Izračunavamo  $\tilde{v}$ :

$$\tilde{v}_1 = \{0.294, 0.294, -0.118, -0.294\}$$

$$\tilde{v}_2 = \{-0.289, 0.289, -0.13, 0.289\}$$

Za  $B$  vrijedi:

$$B = \{4, 4\}$$

Izračunavamo  $W$  uz  $k = 0.08$ :

$$W = \{1.16, 1.16\}$$

Računamo  $\sigma$ :

$$\sigma = \{0.2961, 0.0402, 0.0245, 0.2961\}$$

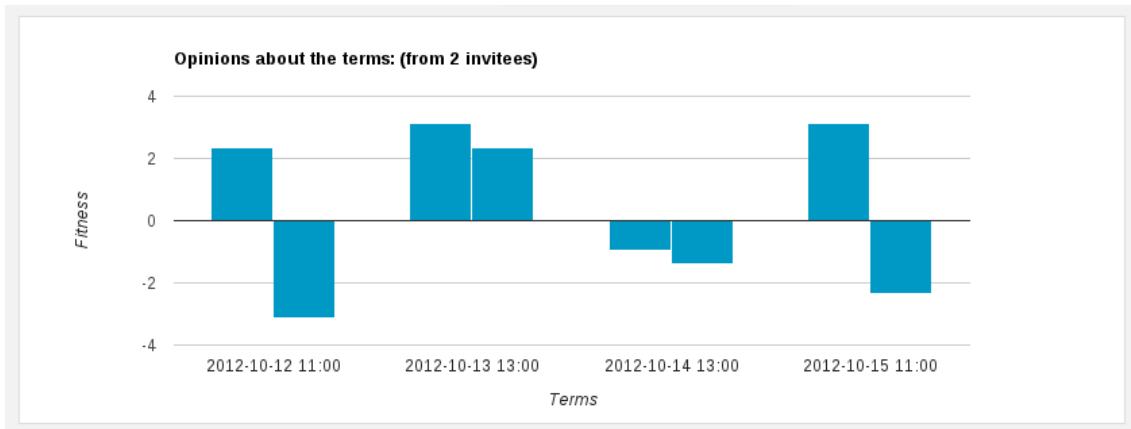
Uz  $q = 0.3$  dobivamo za termine vrijednosti:

$$rezultat = \{-0.1289, 0.284, -0.1342, -0.0486\}$$

Te odabiremo drugi termin budući da ima najveću vrijednosti.

Zanimljivo je primijetiti da četvrti termin ima negativnu vrijednost iako je prosjek mišljenja za taj termin pozitivan. Razlog tome je velika disperzija mišljenja o tom terminu.

Uz navedene parametre, sustav će prikazati graf sa slike 5.9.



**Sl. 5.9.** Grafički prikaz korisničkih mišljenja za primjer 3

## 6. Zaključak

U teorijskom dijelu rada opisane su mogućnosti Google App Engine-a kao platformske usluge oblaka računala. Detaljno je razrađen model planiranja sastanaka koji omogućuje odabir najpovoljnijeg termina za održavanje sastanka. Analizirani su različiti parametri modela.

U praktičnom dijelu rada opisan je primjenSKI program koji je implementiran na Google App Engine oblaku računala. PrimjenSKI program olakšava dogovaranje vremena održavanja sastanaka, te koristi predloženi model. Program je ostvaren na oblaku računala kako bi bio skalabilan, lagan za održavanje, te imao pristup programskim sučeljima App Engine oblaka. Ostvaren je u najnovijim internet tehnologijama te je intuitivan i jednostavan za korištenje. Testiranjem programa pokazano je da se program prilagođava količini korisnika i mogućih termina, nije manipulativan od strane korisnika, te daje rezultate u skladu s očekivanjima.

# LITERATURA

- [1] B. Sosinsky. *Cloud Computing Bible*, Wiley, 2011.
- [2] B. Basham, K. Sierra i B. Bates. *Head First Servlets and JSP*, O'Reilly, 2008.
- [3] *Developing cloud apps: What's different*,  
<http://www.infoworld.com/d/cloud-computing/developing-cloud-apps-whats-different-675>, lipanj 2012.
- [4] E. Robson i E. Freeman. *Head First HTML with CSS & XHTML*, O'Reilly, 2005.
- [5] Google. *Google App Engine*,  
<https://developers.google.com/appengine/>, lipanj 2012.
- [6] G. Martinović. *Framework of Meeting Scheduling in Computer Systems*, Issues in Information Systems. IX, 2; 560-559, 2008.
- [7] M. Čorak. *Razvoj primjenskih programa za sustave oblaka računala*, završni rad, ETF Osijek, 2010.
- [8] *NetBeans*, <http://netbeans.org/>, lipanj 2012.
- [9] Oracle. *The Java EE 6 Tutorial*,  
<http://docs.oracle.com/javaee/6/tutorial/doc/docinfo.html>, lipanj 2012.
- [10] R. Harms i M. Yamartino. *The Economics of the Cloud*, Microsoft, 2010.
- [11] R. W. Douglass, G.A. Hansen i A. Zardecki. *Mesh enhancement*, Imperial College Press, 2005.
- [12] X. Wang, B. Wang i J. Huang. *Cloud computing and its key techniques*, Computer Science and Automation Engineering (CSAE), IEEE International Conference, Xiamen, Kina, 10.-12.06. 2011., pp. 404 – 410
- [13] Y. Jadeja i K. Modi. *Cloud computing - concepts, architecture and challenges*, Computing, Electronics and Electrical Technologies (ICCEET), International Conference, Kherva, Indija, 21.-22.03. 2012., pp. 877 – 880
- [14] Z. Shu-Qing i X. Jie-Bin. *The Improvement of PaaS Platform*, Networking and Distributed Computing (ICNDC), First International Conference, Guangzhou, Kina, 21.-24.10. 2010., pp. 156 – 159

## **Sustav za planiranje sastanaka za Google App Engine oblak računala**

### **SAŽETAK**

Rad opisuje mogućnosti računarstva u oblaku, te posebno pojašnjava prednosti Google App Engine platformske usluge oblaka računala. U radu je razrađen model planiranja sastanaka koji omogućuje odabir najpovoljnijeg termina za održavanje sastanka. U praktičnom dijelu rada implementiran je primjenski program na Google App Engine oblaku računala. Primjenski program olakšava dogovaranje vremena održavanja sastanaka te za odabir najboljeg termina koristi osmišljeni model. Program je ostvaren na oblaku računala kako bi bio skalabilan, lagan za održavanje, te imao pristup programskim sučeljima App Engine oblaka. Program se prilagođava količini korisnika i mogućih termina, te daje rezultate u skladu s očekivanjima.

**Ključne riječi:** Google App Engine, oblak računala, planiranje sastanaka, Platforma kao usluga (PaaS)

## **System for planning meetings for Google App Engine cloud**

### **ABSTRACT**

This paper describes the possibilities of cloud computing, and specifically explains the benefits of the Google App Engine cloud platform. In this paper a model for planning a meetings was developed. The model allows user to select the best term for the meeting. In the practical part of this paper, application was implemented on Google App Engine cloud. Application allows users to choose the best time for the meeting by utilizing described model. Application was implemented on a cloud to be scalable, easy to maintain, and to have access to App Engine programming interfaces. Application adjusts itself to the number of users and number of potential terms, and gives results according to expectations.

**Keywords:** Google App Engine, cloud, planning meetings, Platform as a Service (PaaS)

# ŽIVOTOPIS

Mirela Čorak rođena je 05.05.1989. u Osijeku. Nakon završene prirodoslovno-matematičke gimnazije u Osijeku, 2007. godine upisuje Elektrotehnički fakultet u Osijeku. 2010. godine završava sveučilišni preddiplomski studij računarstva, te upisuje diplomski studij na istom fakultetu, smjer procesno računarstvo.

Pored obveza s fakulteta, Mirela je 2011. godine certificirala svoje znanje programiranja u Javi, te postaje „*Oracle Certified Professional Java SE 6 Programmer*“. Iste godine Mirela polaže online predavanje „*Artificial Intelligence*“ na Stanford University kod profesora Petera Norviga i Sebastiana Thruna s uspješnošću 99.2%, te za to dobiva certifikat. Mirela je također tehnički recenzent knjige „*Napredno programiranje i algoritmi u C-u i C++-u*“ autora Domagoja Kusalića.

Mirela je vješta u sljedećim tehnologijama: Java (Servlets, JSP, JSTL i EL), Google App Engine, HTML 5, CSS 3, JavaScript, jQuery, Linux, SQL i C/C++.

---

Mirela Čorak

# PRILOZI (NA CD-U)

Prilog 1. Tekst rada u \*.docx formatu.

Prilog 2. Tekst rada u \*.pdf formatu.

Prilog 3. Prezentacija \*.pptx formatu.

Prilog 4. Slike korištene u radu.

Prilog 5. Kodovi primjenskog programa.

Prilog 6. Alati, biblioteke i priključci potrebni za rad programa.