

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE
SUSTAVE

Sustav za prosljeđivanje elektroničke pošte

Jurica Zrna
DIPLOMSKI RAD

Zagreb, 2012.

Sadržaj

1..Uvod.....	1
2..Elektronička pošta.....	2
3..Programska potpora.....	3
3.1..Debian.....	3
3.2..Postfix.....	3
3.3..Django.....	3
3.4..MySQL.....	4
3.5..Apache2.....	4
3.6..Ostalo.....	4
4..Postavljanje sustava.....	5
4.1..Instalacija operacijskog sustava.....	5
4.2..Sigurnosna stijena i fail2ban.....	10
4.2.1..Shorewall.....	10
4.2.2..Fail2Ban.....	13
4.3..Instalacija potrebnih paketa.....	13
4.4..MySQL podešavanje.....	14
4.5..Django aplikacija.....	15
4.5.1..Instalacija.....	15
4.5.2..Podešavanje.....	15
4.5.3..Testiranje.....	15
4.6..TSL.....	16
4.6.1..Potrebni ključevi i certifikati.....	16
4.6.2..Kako do certifikata.....	17
4.7..Postfix.....	19
4.7.1..Podešavanje.....	19
4.7.2..Testiranje.....	22
4.8..Apache2.....	22
4.8.1..Podešavanje.....	22
4.8.2..Testiranje.....	23
4.9..Skripte.....	23
4.9.1..Sjednice.....	24
4.9.2..Potvrda adrese elektroničke pošte.....	24
4.9.3..Zaboravljena lozinka.....	24
4.9.4..Enkripcija lozinki.....	24
4.9.5..Skripte.....	25
4.9.6..cron.....	25
5..Izvorni kod.....	26
5.1..Organizacija koda.....	26
5.2..urls.py.....	26
5.3..Administracijski dio.....	26
5.3.1..Modeli.....	26
5.3.2..Django admin.....	27
5.3.3..admin.py.....	28
5.3.4..forms.py.....	28
5.3.5..Pogledi i akcije.....	29
5.3.6..Predlošci.....	31
5.4..Korisnički dio.....	31
5.4.1..Polja.....	32
5.4.2..Forme.....	32

5.4.3..Pogledi.....	33
5.4.4..Predlošci.....	37
5.5..CSS.....	39
6..Korištenje sustava.....	40
6.1..Super korisnik i korisnici.....	40
6.1.1..Spajanje na sustav.....	40
6.1.2..Super korisnik.....	40
6.1.3..Korisnici.....	40
6.1.4..Ustanove.....	43
6.1.5..Studenti.....	44
6.2..Studenti.....	50
6.2.1..Spajanje na sustav.....	50
6.2.2..Promjena podataka.....	50
6.2.3..Zaboravljena lozinka.....	51
7..Zaključak.....	54
8..Literatura.....	55

1. Uvod

Svaki student koji upiše fakultet u Republici Hrvatskoj pri upisu na korištenje dobije adresu elektroničke pošte. Za vrijeme studiranja to je adresa preko koje fakultet u svakom trenutku može stupiti u kontakt sa studentom.

Međutim prilikom završetka studija ta se adresa briše. Osim što je to nepraktično za studente, fakultet također gubi kontakt sa svojim bivšim studentima.

Da bi se doskočilo tim problemima treba stvoriti sustav za upravljanje elektroničkom poštom. Ovaj rad prikazuje jedno takvo rješenje. Glavna mu je zadaća bivšim studentima pružati uslugu preusmjeravanja elektroničke pošte s adrese na domeni fakulteta, na bilo koju adresu elektroničke pošte studenta.

Osim adrese elektroničke pošte, bivši student je u mogućnosti ostaviti i ostale podatke za kontakt, adresu te broj telefona.

Za ostvarenje sustava za upravljanje elektroničkom poštom koriste se isključivo besplatno dostupni alati (engl. open source tools). Na taj način smanjuje se cijena sustava i povećava njegova fleksibilnost. Cilj je postaviti sustav koji je pristupačan korisnicima, no istodobno siguran, pouzdan te jednostavan za nadogradnju.

U poglavlju 1 napisan je kratak uvod u diplomski rad.

Poglavlje 2 opisuje usluge elektroničke pošte.

Programska potpora potrebna za ostvarenje sustava za upravljanje elektroničkom poštom opisana je u poglavlju 3.

U poglavlju 4 prikazan je postupak izgradnje sustava za upravljanje elektroničkom poštom. Peto poglavlje pojašnjava izvorni kod Web aplikacije sustava koja je izgrađena u sklopu ovog rada.

U poglavlju 6 opisan je način na koji se sustav koristi.

Rad završava zaključkom u 7. poglavlju.

2. Elektronička pošta

Elektronička pošta je najstariji način za izmjenu poruka na Internetu. Datira još od ranih sedamdesetih godina dvadesetog stoljeća, čime je starija od samog Interneta i jedan o njegovih pokretača. Poruke se razmjenjuju takozvanim "spremi i proslijedi" načinom, što znači da svaki poslužitelj sprema poruke i prosljeđuje ih kad je sljedeći poslužitelj dostupan, čime se izbjegava potreba za stalnom dostupnosti svih poslužitelja i korisnika, te oni moraju biti istovremeno dostupni samo djelić vremena koji je potreban za razmjenu poruka.

Poruka koja se prosljeđuje sastoji se od dva dijela, zaglavlja i tijela. U zaglavlju se nalaze različita polja za informacije: primatelj, pošiljatelj, naslov, vrijeme i ostale informacije o poruci. U tijelu se nalazi sama poruka, obično tekst, vrlo slično poruci koju šaljemo u kuverti običnom poštom.

Razmjenjivanje između poslužitelja vrši se SMTP protokolom, dok klijenti obično pristupaju protokolima POP i IMAP, ili nekim od vlasničkih protokola. U današnje vrijeme uobičajeno je da se elektroničkoj pošti može pristupiti i preko Web sučelja.

Kako je elektronička pošta stari sustav, tako za sobom ima niz problema. Sama po sebi nema nikakav način provjere pošiljatelja, cijeli sustav za razmjenu vrši se neenkriptiranim porukama, i ono što je danas možda najpoznatiji problem, elektronička pošta je zatrpana neželjenim porukama, takozvanim *spamom*. Neki od tih problema riješeni su, razmjena se danas uglavnom enkriptira, razvijene su metode za osiguranje povjerljivosti i neporecivosti poput *PGP*-a. Međutim *spam* i dalje čini ogroman dio razmjenjenih poruka.

3. Programska potpora

U ovom poglavlju bit će predstavljena programska potpora koja je korištena u stvaranju sustava za prosljeđivanje elektroničke pošte.

3.1. Debian

Debian je *GNU/Linux* operacijski sustav. Za razliku od popularnih komercijalnih, no i mnogih besplatnih operacijskih sustava, *Debian* nema nikakvu komercijalnu kompaniju iza sebe, održavan je i razvijan isključivo od dobrovoljaca iz zajednice. Jedan je od najstarijih *GNU/Linux* sustava i datira iz 1993. godine.

Debian je odabran zbog nekoliko razloga. Kako je razvijan od zajednice i već ima iza sebe velik broj inačica i dosta staža, vjerojatnost da će "nestati" vrlo je mala. Iako nema službene podrške od samog *Debian* projekta, brojni volonteri uvijek su spremni pomoći na Internetu, što na Web stranicama, što na *IRC*-u, a postoje i brojne tvrtke koje pružaju komercijalnu podršku. *Debian* koristi *apt* sustav za instalaciju, deinstalaciju i nadogradnju paketa. Time je znatno olakšana briga o sustavu, jer se sam sustav brine o tome da svi instalirani dijelovi međusobno funkcioniraju i pruža laganu i bezbolnu nadogradnju. Još jedan od razloga je što je *Debian* u svijetu jedan od najpopularnijih *GNU/Linux* sustava za poslužitelje, samim time postoji znatna dokumentacija baš na tom području.

U ovom sustavu korištena je verzija "wheezy". U trenutku pisanja ovog rada to je test verzija *Debian*a, međutim očekuje se da će u sljedećih par mjeseci stasati u stabilnu verziju 7.0. Nakon toga slijede dvije godine u kojima će sustav biti potpuno podržan kao stabilna verzija, te još godinu dana u kojima bi sustav trebalo nadograditi na novu verziju.

3.2. Postfix

Postfix je poslužitelj elektroničke pošte. Originalno razvijan od *IBM*-a, danas je besplatan i otvoren. To je trenutno najviše ispitivan poslužitelj elektroničke pošte, pogodan za posluživanje malih sustava poput osobnog poslužitelja do ogromnih sustava koji poslužuju desetke tisuća korisnika.

U ovom sustavu korištena je verzija 2.9.3, koja dolazi s *Debian* sustavom.

Podešavanje se vrši modifikacijom dviju datoteka:

```
/etc/postfix/main.cf  
/etc/postfix/master.cf
```

Datoteka *main.cf* sadrži parametre poslužitelja, *master.cf* upravlja pojedinim dijelovima poslužitelja, takozvanim *daemonima* (uslužnim programima, servisima koji rade u "pozadini, bez izravne interakcije s korisnikom).

3.3. Django

Django je radno okruženje za razvoj Web programa (stranica) izgrađen u programskom jeziku *python*. Služi brzom razvoju dinamičkih Web stranica uz što manje ulaganje vremena na radnje koje se ponavljaju i pozadinski kod.

U sustavu se koristi zadnja stabilna verzija 1.4. (u trenutku izrade ovog rada)

3.4. MySQL

MySQL je vjerojatno najpoznatija i najrasprostranjenija relacijska baza podataka na Webu. U ovom sustavu služi kao centralno mjesto na kojem se drže podatci za poslužitelje.

Podešavanje baze podataka vrši se stvaranjem datoteka u mapi:

```
/etc/mysql/conf.d/
```

U sustavu se koristi verzija 5.5.24.

3.5. Apache2

Jedan od najrasprostranjenijih i najpopularnijih Web poslužitelja na Internetu. Modularnog dizajna i provjerene pouzdanosti.

U sustavu će služiti posluživanju *Django* Web stranica pomoću *mod_wsgi* modula.

Podešavanje se ostvaruje stvaranjem datoteka u mapi:

```
/etc/apache2/conf.d/
```

Verzija koja se koristi je 2.2.22.

3.6. Ostalo

Shorewall

Shorewall je skripta za podešavanje *Linux IPtables* sigurnosne stijene. *Shorewall* sam po sebi nije sigurnosna stijena, već se on pokreće kao skripta, podešava *IPtables* te se potom gasi.

Koristiti će se za ograničavanje pristupa poslužitelju samo na one resurse koje mi želimo učiniti dostupnima.

Za podešavanje se koristi niz datoteka u mapi:

```
/etc/shorewall
```

Sustav ima instaliranu verziju 4.5.3.

Fail2ban

Uobičajeno je da su poslužitelji neprestano povezani na Internet. Ta činjenica omogućava potencijalnim napadačima veliki vremenski period za analizu i napad na sustav. *Fail2ban* analizira logove od različitih procesa, identificira neke od napada i onemogućava pristup napadaču.

U sustavu će se koristiti za sprečavanje napada "grubom silom" (engl. *brute force*) na *SSH* poslužitelj.

Potrebu za ovime najbolje ilustrira primjer testnog poslužitelja za ovaj rad, koji je postao meta prvog napada nepunih 24 sata nakon pokretanja, bez da je bilo gdje bila objavljena adresa poslužitelja.

Podešavanje se vrši u datoteci:

```
/etc/fail2ban/jail.conf
```

Koristi se verzija 0.8.6.

4. Postavljanje sustava

Slijedi opis postupka postavljanja sustava. Od nabavke programa do kompletnog podešavanja svakog instaliranog paketa. Slijedeći ovo poglavlje moguće je replicirati sustav na drugom računalu.

4.1. Instalacija operacijskog sustava

Nabavka instalacijskih datoteka

Mediji za instalaciju *Debian* operacijskog sustava distribuiraju se u različitim verzijama, od minijturnih verzija namijenjenih instalaciji preko Interneta, do kompletnog sustava za instalaciju koji trenutno broji 8 DVD medija.

Moja preporuka je *netinst* verzija, veličine ispod 200Mb, koja prilikom instalacije preuzima potrebne datoteke s Interneta.

Mediji su u vrijeme pisanja ovog rada dostupni na:

```
http://www.debian.org/devel/debian-installer/
```

Dok će nakon izlaska verzije 7.0 biti dostupni na:

```
http://www.debian.org/distrib/
```

Na Web stranici je moguće izabrati različite protokole za preuzimanje, arhitekture i veličine medija.

U ovom radu koristit će se *netinst amd64* verzija, dostupna na:

```
http://cdimage.debian.org/cdimage/wheezy_di_alpha1/amd64/iso-cd/debian-wheezy-DI-a1-amd64-netinst.iso
```

Izrada instalacijskog medija

Instalacijski mediji dolaze u *iso* formatu koji se može snimiti na CD/DVD pomoću većine programa za snimanje u svim operacijskim sustavima.

Ukoliko se za izradu medija koristiti *GNU/Linux* moguće je vrlo lako izraditi instalacijski medij na USB memoriji. To se radi tako što se preuzme instalacijski medij s Interneta, priključi USB memoriju, te izvrši sljedeće naredbe:

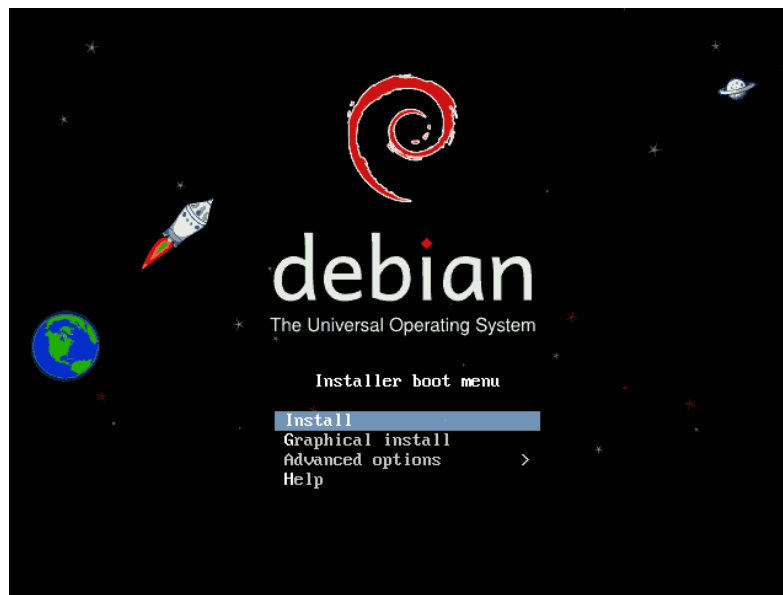
```
# cat debian.iso > /dev/sdX  
# sync
```

Pri čemu je */dev/sdX* lokacija USB memorije.

Instalacija operacijskog sustava

Nakon što je spreman medij, potrebno ga je ubaciti u računalo, pokrenuti ga, te ući u BIOS. To se obično radi pritiskom na tipku *Del* prilikom pokretanja računala. U BIOSu je potrebno odabrati instalacijski medij kao početni (engl. *boot-*) medij te snimiti promjene i ponovo pokrenuti računalo.

Pri pokretanju pojaviti će se slijedeći izbornik:



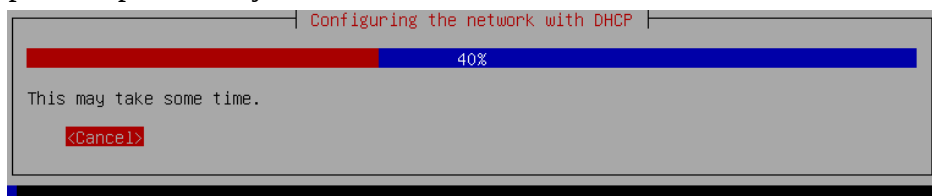
Odabrati *install* i pričekati. Slijedi izbor jezika. Iako je ponuđen *hrvatski*, bolji izbor je *engleski*. Razlog je taj što je sva pomoć na Internetu gotovo isključivo na engleskom jeziku pa je na taj način mnogo lakše pronaći i primijeniti pomoć ukoliko dođe do problema.

Nakon jezika slijedi lokacija, odabrati po želji. Moj izbor je *Hrvatska*, pod *Other*, pa *Europe*.

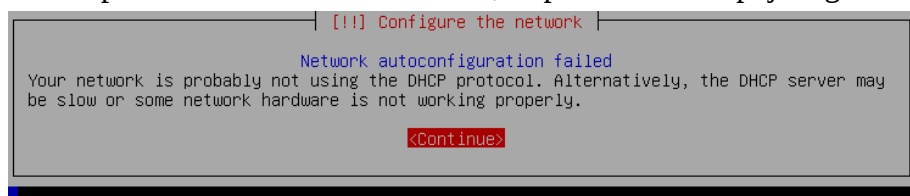
Slijedeći korak je izbor lokalnih postavki, tu je izbor prilično mali i svodi se na nekoliko verzija engleskog, tako da je moj izbor podrazumijevani, *United States*.

Slijedi izbor rasporeda tipki na tipkovnici. Ja sam izabrao *hrvatski*.

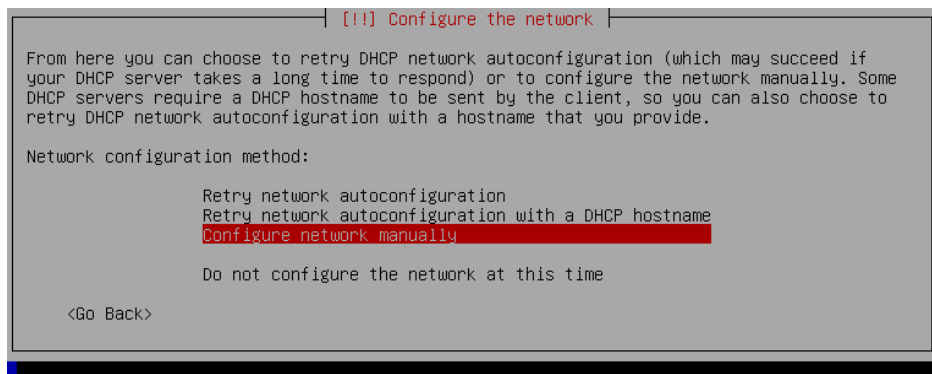
Zatim se pokreće podešavanje mreže.



Ukoliko se koristi *DHCP* za podešavanje mreže, postavke će biti automatski prepoznate i postavljene. U suprotnom treba odabrati *Cancel*, ili pričekati da se pojavi greška.



Nakon pritiska na *Continue*, pojavljuje se izbornik za ručno podešavanje mreže.



Ovdje je moguće podesiti postavke mreže, IP adresu, *subnet* masku, *gateway*.

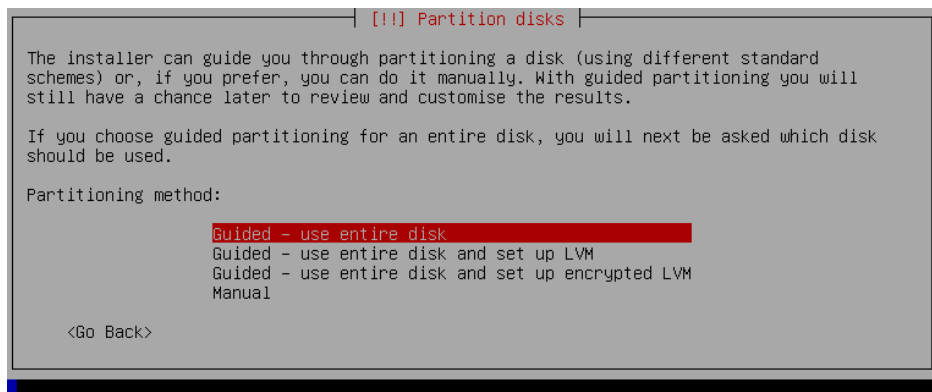
Nakon što je mreža uspješno postavljena, potrebno je unijeti naziv računala (engl. *hostname*), u mom slučaju to je "dipl1916". Zatim domenu, dio adrese računala na Internetu bez imena računala.

Slijedi unos lozinki za *root* korisnika, ovdje će se koristiti "root2012", zatim ime korisnika, "website" i lozinka, "web2012".

Naravno, lozinke na poslužitelju koji je otvoren prema Internetu moraju biti jedinstvene i dobre, barem 8 znakova dužine, velika i mala slova, brojke i specijalni znakovi.

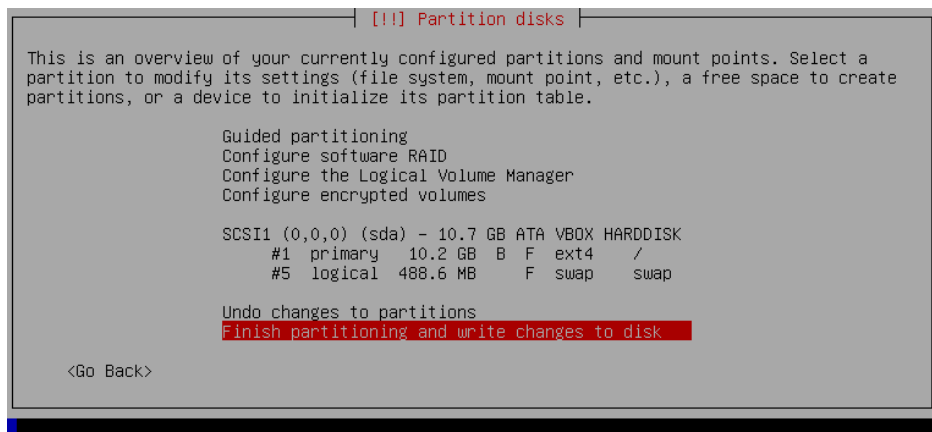
Korisnik je nazvan "website" jer će se u njegovom korisničkom računu izvoditi Web stranica koja će se posluživati.

Slijedi particioniranje diska.



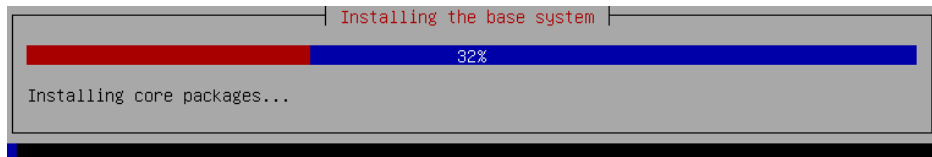
Odabrati po potrebi, vođeno particioniranje je generalno dobar izbor, enkriptiranje diska na poslužiteljima nije potrebno jer ne bi trebali biti dostupni neovlaštenom osoblju, a samo kriptiranje zahtjeva procesorsko vrijeme.

Prikazati će se izvješće o tome kako će se particionirati disk.



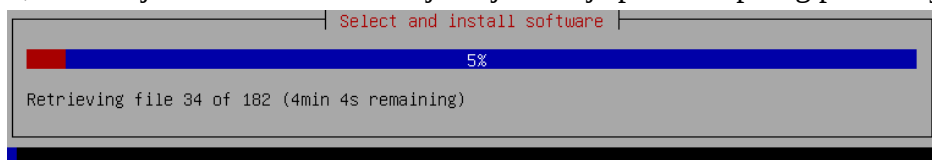
Odaberi završetak i spremanje promjena.

Slijedi instalacija glavnog sustava.

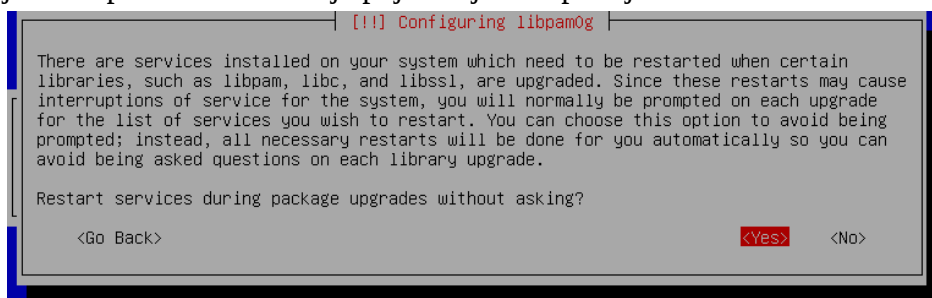


Nakon čega treba odabrati *mirror*. Najbolje je odabrati nešto što se fizički nalazi blizu poslužitelju, u mom slučaju to je bio CARNet.

Slijedi instalacija paketa. Kod instalacije se automatski preuzimaju najnovije verzije paketa s Interneta, tako da je instalirani sustav najnovije verzije prilikom prvog pokretanja.



Moguće je da se prilikom instalacije pojavi sljedeće pitanje:

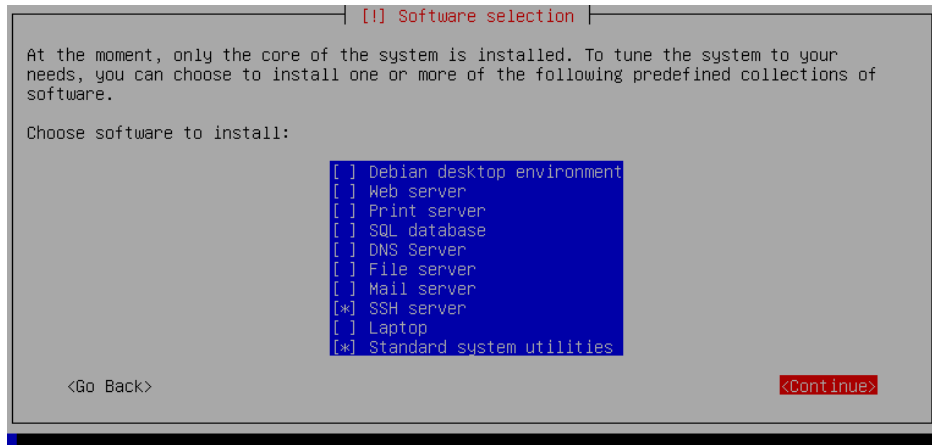


Odaberi Yes, jer još uvijek traje instalacija sustava i svi se servisi mogu ponovo pokrenuti bez straha od prekida usluge. Isto se može dogoditi prilikom nadogradnje sustava, tada je potrebno puno pažljivije pratiti koji servisi se ponovo pokreću, jer je moguće da dođe do prekida usluge ukoliko se prekine neki od servisa koji koristimo (na primjer, ukoliko se na poslužitelj spajamo preko SSH veze, a ponovo pokrenemo SSH poslužitelj, izgubiti će se

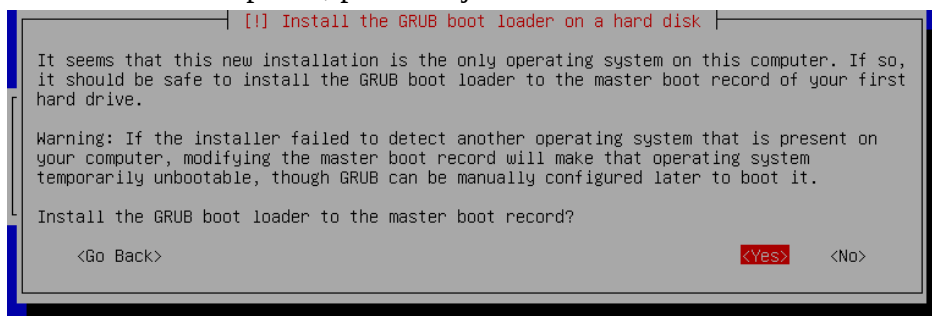
veza na poslužitelj).

Pojaviti će se upitnik za "popularity contest", koji javlja razvojnom timu *Debiana* koje pakete koristi sustav, te se ti podatci koriste kod odlučivanja koji paketi se uvrštavaju u sljedeće verzije *Debiana*. Odaberi po želji.

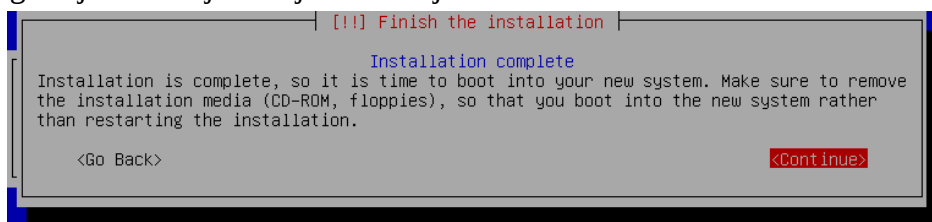
Slijedi odabir dodatnih paketa koji će se instalirati, potrebno je odabrati "SSH server" i "Standard system utilities". "SSH server" je siguran način za udaljeni pristup računalu, dok je "Standard system utilities" skup korisnih alata.



Nakon što su instalirani svi paketi, potrebno je instalirati *GRUB* boot loader.



Poslije toga slijedi obavijest da je instalacija završena.



Potrebno je maknuti instalacijski medij i pritisnuti *Continue* za ponovno pokretanje poslužitelja.

Nakon podizanja sustava, prikazuje se upitnik za prijavu (engl. *login prompt*).

4.2. Sigurnosna stijena i *fail2ban*

Nakon instalacije, poslužitelj je spojen na mrežu bez ikakve prepreke. Da bi se spriječio neželjeni pristup i kontrolirao željeni pristup, potrebna je sigurnosna stijena. U ovom slučaju to će biti *IPtables* podešen pomoću *shorewalla*.

Da bi se izvršila bilo kakva naredba, potrebno je prijaviti se u sustav unosom korisničkog imena i lozinke.

4.2.1. Shorewall

Instalacija

Prije svake instalacije paketa dobro je izvršiti naredbu:

```
# apt-get update
# apt-get upgrade
```

Time se Debian ažurira na najnoviju verziju. Kako je naš sustav upravo instaliran to nije potrebno.

Pokrenemo naredbu:

```
# apt-get install shorewall
```

Poslužitelj će nas obavijestiti o promjenama:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  shorewall-core
Suggested packages:
  shorewall-doc
The following NEW packages will be installed:
  shorewall shorewall-core
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 728 kB of archives.
After this operation, 1,886 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Pritisnuti *enter* za potvrdu instalacije.

Potrebni paketi bit će dohvaćeni s Interneta i instalirani na poslužitelj.

```
Get:1 http://ftp.hr.debian.org/debian/ wheezy/main shorewall-core all 4.5.5.1-1 [48.5 kB]
Get:2 http://ftp.hr.debian.org/debian/ wheezy/main shorewall all 4.5.5.1-1 [679 kB]
Fetched 728 kB in 5s (140 kB/s)
Preconfiguring packages ...
Selecting previously unselected package shorewall-core.
(Reading database ... 41549 files and directories currently installed.)
Unpacking shorewall-core (from ../shorewall-core_4.5.5.1-1_all.deb) ...
Selecting previously unselected package shorewall.
Unpacking shorewall (from ../shorewall_4.5.5.1-1_all.deb) ...
Processing triggers for man-db ...
Setting up shorewall-core (4.5.5.1-1) ...
Setting up shorewall (4.5.5.1-1) ...
```

Podešavanje

Podešavanje *shorewalla* sastoji se od nekoliko koraka. Prvi je definicija zona.

Otvorimo datoteku:

```
# nano /etc/shorewall/zones
```

Zone su područja na mreži. Svaka zona odgovara jednom mrežnom sučelju. U ovom slučaju postoje samo dvije zone, računalo i sve ostalo, no u složenijim sustavima često postoji više zona poput Interneta, lokalne mreže, demilitarizirane zone.

U datoteku dodamo:

#ZONE	TYPE	OPTIONS	IN OPTIONS	OUT OPTIONS
#				
fw	firewall			
net	ipv4			

Definirane su dvije zone, prva je specijalna zona *fw* koja označava računalo na kojem se nalazi *shorewall*, druga zona je *net* i koristi IPv4 adrese.

Sljedeći korak je definicija sučelja za svaku zonu. Otvori se datoteka:

```
# nano /etc/shorewall/interfaces
```

I doda:

```
FORMAT 2
#####
#
#ZONE      INTERFACE      OPTIONS
net        eth0            dhcp, tcpflags, logmartians, nosmurfs
```

Ovdje se nalazi par opcija uz zonu, one služe filtriranju nekih neželjenih paketa s Interneta. Nakon što su definirana sučelja, potrebno je definirati generalna pravila za pojedine pakete, takozvanu politiku sigurnosne stijene.

Ona se nalazi u datoteci:

```
# nano /etc/shorewall/policy
```

U datoteku se upiše:

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT: BURST	CONNLIMIT: MASK
#					
\$FW	net	ACCEPT			
net	\$FW	DROP	info		
net	all	DROP	info		
# The FOLLOWING POLICY MUST BE LAST					
all	all	REJECT	info		

Ovakva politika govori sljedeće:

- svi paketi od računala prema mreži se prihvaćaju
- svi paketi od mreže prema računalu se odbacuju
- svi paketi od mreže prema nekom drugom odredištu se odbacuju
- svi paketi se odbijaju (odbacuju uz obavještanje izvora)

Ovo je prilično striktna politika i sprječava sav promet s mreže na računalo, da bi poslužitelj mogao funkcionirati, do njega moraju dolaziti zahtjevi s mreže, za to služe pravila, to su zapravo iznimke od politike sigurnosne stijene. Ona se nalaze u:

```
# nano /etc/shorewall/rules
```

A datoteka za početak sadržava:

#ACTION ORIGINAL \ HEADERS #	SOURCE RATE SWITCH	DEST USER/ GROUP	PROTO MARK	DEST CONNLIMIT	SOURCE PORT(S)	TIME
#SECTION ALL						
#SECTION ESTABLISHED						
#SECTION RELATED						
SECTION NEW						
SSH/ACCEPT	net	\$FW				
Ping/ACCEPT	net	\$FW				
# Permit all ICMP traffic FROM the firewall TO the net zone						
ACCEPT	\$FW	net			icmp	

Treba primijetiti da su prve dvije linije zapravo jedna linija spojena sa znakom \.

Ova pravila omogućavaju *SSH* pristup poslužitelju, dopuštaju *ping* s mreže i *icmp* pakete prema mreži.

Ovo je izgled datoteke na početku, kasnije, kako će biti postavljene različite usluge, tako će biti i dodana pravila za njih.

Da bi se mogao pokrenuti *shorewall*, potrebno je u datoteci:

```
# nano /etc/default/shorewall
```

Promijeniti:

```
startup=0
```

U:

```
startup=1
```

Za provjeru sintakse pokrene se:

```
# shorewall check
```

Te je moguće pokrenuti sigurnosnu stijenu.

UPOZORENJE:

Ukoliko se radi na udaljenom računalu bitno je biti siguran da su sve postavke ispravne jer je moguće "zaključati" se iz računala ukoliko neka od postavki nije dobro postavljena!

Nakon što smo se uvjerali da su sve postavke ispravne, pokrenuti:

```
# /etc/init.d/shorewall start
```

Time je sigurnosna stijena pokrenuta.

Problemi

Ukoliko dođe do problema koji upućuju na sigurnosnu stijenu, moguće ju je privremeno isključiti naredbom:

```
# /sbin/shorewall clear
```

Nakon testiranja ključno je opet pokrenuti:

```
# /etc/init.d/shorewall start
```

U suprotnom, poslužitelj će biti potpuno otvoren prema mreži.

4.2.2. Fail2Ban

Instalacija

Program se instalira pokretanjem naredbe:

```
# apt-get install fail2ban
```

Podešavanje

Podešavanje se vrši kopiranjem datoteke koja dolazi s programom i modifikacijom iste:

```
# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
# nano /etc/fail2ban/jail.local
```

U datoteci izmijeniti:

```
ignoreip = 127.0.0.1/8
bantime = 600
maxretry = 3
```

Prva opcija su razmakom odijeljene mreže koje se nikad ne blokiraju. Ovdje je dobro staviti sve adrese kojima vjerujemo, tako da nikad ne budemo zaključani iz poslužitelja.

Vrijeme zabrane u sekundama nalazi se u *bantime*, a broj neuspjelih pokušaja prije zabrane u *maxretry*.

```
destemail = admin@domena.com
```

Adresa na koju se šalju poruke s izvješćima o blokiranim adresama.

```
banaction = shorewall
```

Način na koji se vrše zabrane.

```
action = %(action_mwl)s
```

Akcija koja se vrši prilikom blokiranja. U ovom slučaju to zabrana pristupa i slanje izvješća s relevantnim linijama loga i informacijama o napadaču.

Ponovo pokrenemo program:

```
# /etc/init.d/fail2ban restart
```

4.3. Instalacija potrebnih paketa

Nakon što je poslužitelj osiguran od neželjenog pristupa, vrijeme je za postavljanje sustava. Prvi korak je instalacija potrebnih paketa.

Izvrši se naredba:

```
# apt-get install postfix postfix-mysql mysql-server mutt tumgreyspf python-django python-mysqldb python-bcrypt python-xml python-dev libapache2-mod-wsgi
```

Ovime su instalirani svi paketi potrebni za funkcioniranje poslužitelja.

```
postfix postfix-mysql mutt tumgreyspf
```

Poslužitelj za elektroničku poštu, klijent za pregled elektroničke pošte, alat za sive liste.

```
mysql-server
```

MySQL baza podataka

```
python-django python-mysqldb python-bcrypt python-xml python-dev
```

Django Web framework i prateći alati za pristup bazi podataka i enkripciju lozinki.

```
libapache2-mod-wsgi
```


Modul za *apache2* Web poslužitelj za posluživanje *python* koda.

Za vrijeme instalacije pojavit će se upitnik za *postfix*, odabrati "Internet Site" i upisati domenu za koju poslužitelj prihvaća elektroničku poštu.

Također, bit će zatražena lozinka za *root* korisnika baze podataka, u ovom radu koristiti će se "mysql2012", u realnom sustavu vrijede ista pravila kao i kod instalacije operacijskog sustava.

4.4. MySQL podešavanje

Podešavanje se vrši stvaranjem datoteke:

```
# nano /etc/mysql/conf.d/unicode.cnf
```

Dodati:

```
[mysqld]
#
# * Default to unicode
#
collation-server = utf8_general_ci
init-connect='SET NAMES utf8'
character-set-server = utf8
```

Na taj način baza će koristiti *unicode* znakove ukoliko nije drugačije navedeno. To omogućava spremanje znakova iz gotovo bilo kojeg jezika.

Kao i poslije svakog podešavanja, potrebno je ponovo pokrenuti servis:

```
# /etc/init.d/mysql restart
```

Poslužitelj će posluživati dva servisa, elektroničku poštu i Web stranice, oba servisa trebaju pristup bazi podataka. Loša je praksa bilo kojem servisu dati *root* pristup bazi podataka, jer ukoliko dođe do kompromisa servisa, cijela baza je otvorena napadaču. Stoga je potrebno napraviti korisnika baze podataka za svaki servis, s ovlastima koje servis zahtjeva.

No prvi korak je stvoriti bazu koju će poslužitelj koristiti.

```
# mysqladmin -u root -p create mailfwd
```

Pritom je potrebno je unijeti lozinku.

Spajanje na bazu vrši se naredbom:

```
# mysql -u root -p mailfwd
```

Servisi koji se poslužuju imaju različite zahtjeve, Web poslužitelj mora biti u mogućnosti čitati i modificirati podatke u bazi, dok poslužitelj elektroničke pošte treba samo čitati podatke iz baze. Stoga će biti stvorena dva korisnika, *website* i *postfix* s odgovarajućim ovlastima:

```
GRANT SELECT ON mailfwd.*
TO 'postfix'@'127.0.0.1'
IDENTIFIED BY 'post2012';
GRANT SELECT ON mailfwd.*
TO 'postfix'@'localhost'
IDENTIFIED BY 'post2012';
GRANT ALL ON mailfwd.*
TO 'website'@'127.0.0.1'
IDENTIFIED BY 'website2012';
GRANT ALL ON mailfwd.*
TO 'website'@'localhost'
IDENTIFIED BY 'website2012';
```

Kao i uvijek, potrebno je koristiti jedinstvene i razumne lozinke za svakog korisnika. Svaki korisnik javlja se dva puta u ovom primjeru jer baza različito interpretira pristup s *localhost* i 127.0.0.1.

4.5. Django aplikacija

Instalacija, podešavanje i testiranje *Django* aplikacije (program napisan za sustav za upravljanje elektroničkom poštom u *python/Django*).

4.5.1. Instalacija

Da bi se instalirala aplikacija na poslužitelj, potrebno je arhivu s datotekama otpakirati u mapu:

```
/home/website/mailfwd
```

4.5.2. Podešavanje

Nakon što su datoteke smještene na odgovarajuće mjesto na disku, potrebno je izvršiti podešavanje aplikacije.

To se vrši u datoteci:

```
$ nano /home/website/mailfwd/mailfwd/settings.py
```

Za početak je potrebno popuniti neka polja:

U polju *DATABASES*:

- *NAME* je ime baze podataka
- *USER* je ime korisnika baze podataka ("website")
- *PASSWORD* je lozinka baze podataka ("website2012")

Polje *TIME_ZONE* definira vremensku zonu u kojoj se nalazi poslužitelj.

U polju *SERVER_ADDRESS* nalazi se adresa poslužitelja.

Polje *CONTACT_MAIL* sadržava kontakt adresu poslužitelja.

I polje *NO_REPLAY_MAIL* je adresa za koju se primljena elektronička pošta odbacuje. Ona služi za slanje automatskih poruka, čime se sprječava punjenje diska automatskim odgovorima s adresa na koje se šalju automatske poruke.

Nakon što je postavljena aplikacija, potrebno je stvoriti tablice u bazi podataka. Da bi se to napravilo potrebno je otići u mapu:

```
$ cd /home/website/mailfwd
```

Te pokrenuti naredbu:

```
$ python manage.py syncdb
```

Time će se stvoriti sve tablice potrebne za aplikaciju. U procesu biti će postavljen upit za stvaranje super korisnika Web stranice. Potrebno je odabrati ime po želji (root, admin, administrator) i popuniti upite koji slijede.

4.5.3. Testiranje

Django dolazi sa testnim poslužiteljem. Njegova namjena je isključivo testiranje i nikada se ne smije koristiti za posluživanje Web stranica korisnicima.

Da bi ga pokrenuli, potrebno je otići u:

```
$ cd /home/website/mailfwd
```

Te pokrenuti sljedeću naredbu:

```
$ python manage.py runserver <ip_adresa_poslužitelja>:8000
```

Da bi bilo moguće pristupiti poslužitelju, potrebno je omogućiti pristup u sigurnosnoj stijeni:

```
# nano /etc/shorewall/rules
```

Potrebno je dodati linije:

```
# Django test server
ACCEPT          net          $FW      tcp      8000
```

Te ponovo pokrenuti sigurnosnu stijenu:

```
# /etc/init.d/shorewall restart
```

Nakon čega je moguće pristupiti poslužitelju na adresi:

```
http://<ip_adresa_poslužitelja>:8000
```

Kod testiranja, potrebno je dodati unos za barem jednog studenta, za kasnije testiranje poslužitelja elektroničke pošte.

Nakon testiranja, potrebno je maknuti unos iz sigurnosne stijene i ponovo ju pokrenuti.

4.6. TSL

S razvojem Interneta, te njegovom sve širom primjenom, došlo je do novih potreba u sustavu. Jedna od tih potreba bila je i enkripcija. Riješenje tog problema pronađeno je u SSL-u, a kasnije TLS-u.

TLS je skraćenica za *Transport Layer Security*, a naziv je za protokol koji nudi sigurni prijenos podataka na Internetu. Koristi se asimetrični algoritam za razmjenu ključeva i simetrični algoritam za razmjenu podataka.

Enkripcija je riješila samo jedan od problema, onaj sigurnog prijenosa podataka. Slijedeći veliki problem je identifikacija, naime svatko na Internetu može se predstaviti kao bilo tko. Da bi se doskočilo tome, stvoreni su certifikati.

Certifikat je potpisani javni ključ, on je garancija (od potpisnika) da taj ključ pripada određenoj fizičkoj ili pravnoj osobi. Da bi se vjerovalo nekom certifikatu, potrebno je vjerovati onome tko ga izdaje.

Stoga su nastali *Certificate Authority*, to su tijela od povjerenja čija je zadaća provjeriti identitet onoga tko traži certifikat i potpisati njegov ključ (stvoriti certifikat). Uobičajeno je da su to privatne tvrtke koje za određenu cijenu izdaju certifikate.

Međutim, kao što će biti prikazano kasnije, postoje i alternativni načini za dobivanje certifikata.

4.6.1. Potrebni ključevi i certifikati

Sustav će koristiti dva ključa, jedan za poslužitelja elektroničke pošte na:

```
mail.domena.com
```

I drugi, za Web poslužitelj, na adresi:

```
www.domena.com
```

4.6.2. Kako do certifikata

Postoji nekoliko načina za dobivanje valjanog certifikata, svaki od njih ima svoje prednosti i mane.

Plaćeni certifikat

Prednosti:

- certifikat je automatski prihvaćen od gotovo svih programa za pregled Weba i elektroničke pošte
- to je jedina realna opcija ukoliko je Web stranica komercijalnog karaktera

Nedostaci:

- cijena, certifikati se plaćaju i moraju se obnavljati (i opet plaćati) svake godine ili par godina

Da bi kupili certifikat, potrebno je generirati zahtjev za certifikat i privatni ključ. *Debian* nudi skriptu koja to radi automatski.

Naredba je:

```
# /usr/lib/ssl/misc/CA.pl -newreq
```

Usljedit će niz pitanja:

```
Generating a 2048 bit RSA private key
..+++
.....+++
writing new private key to 'newkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HR
State or Province Name (full name) [Some-State]:Grad Zagreb
Locality Name (eg, city) []:Zagreb
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Dipl1916
Organizational Unit Name (eg, section) []:web server
Common Name (e.g. server FQDN or YOUR name) []:www.domena.com
Email Address []:webmaster@domena.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request is in newreq.pem, private key is in newkey.pem
```

Prvo se generira privatni ključ. Za njega je potrebno upisati lozinku. Zatim je potrebno unijeti podatke o certifikatu, ključno je u polje *Common Name* unijeti adresu servera za koji se zahtjeva certifikat.

Ovdje je primjer za Web poslužitelj, postupak za poslužitelj elektroničke pošte je identičan

uz promjenu polja *Organizational Unit*, *Common Name* i *Email Address* u adekvatne vrijednosti.

Ovaj proces generira dvije datoteke, *newreq.pem* i *newkey.pem*, koje je dobro preimenovati. Dobra praksa je preimenovati datoteke u oblik *<domena>.req* i *<domena>.key*.

Nakon što je zahtjev spreman, potrebno je spojiti se na Web stranicu *Certificate Authority* koji je odabran i slijediti upute. U nekom trenutku bit će potrebno priložiti zahtjev za certifikat.

Nakon što proces završi, CA će vratiti potpisani certifikat spreman za korištenje na Web stranici.

Besplatni Certifikat

Prednosti:

- manje posla
- besplatni

Nedostaci:

- nisu uključeni u sve preglednike i klijente za elektroničku poštu

Postupak je identičan kao i za plaćeni certifikat, jedino što potpisivanje certifikata prepuštamo besplatnom CA.

U ovom trenutku jedini besplatni CA nalazi se na:

`www.cacert.org`

Vlastiti Certificate Authority

Prednosti:

- besplatni
- jednostavno korištenje unutar organizacije

Nedostaci:

- preglednici ne prepoznaju certifikate automatski i javljaju greške
- uči korisnike da prihvaćaju certifikate bez razmišljanja

Da bismo postali CA, potrebno je pokrenuti naredbu:

```
# /usr/lib/ssl/misc/CA.pl -newca
```

Time će se stvoriti nova mapa imenom *DemoCA* sa certifikatom. Pitanja su slična kao i kod stvaranja zahtjeva za certifikat. Polje *Common Name* dobro je završiti s "CA" kako bi uvijek bilo jednostavno prepoznati taj certifikat.

Nakon što je stvoren CA, potrebno je generirati zahtjev za certifikat na isti način kao i kod plaćenih CA, te zatim potpisati taj certifikat sa naredbom:

```
# /usr/lib/ssl/misc/CA.pl -sign
```

Tako potpisani certifikat možemo koristiti na poslužitelju. Ovu naredbu potrebno je

pokrenuti prije preimenovanja datoteka.

4.7. Postfix

Postfix prepoznaje tri vrste domena, lokalne, virtualne i virtualne domene za prosljeđivanje.

Svaki korisnik na poslužitelju automatski ima adresu za svaku domenu za koju poslužitelj prihvaća elektroničku poštu. Pošta se sprema u:

```
/var/mail/korisničko_ime
```

Za virtualne domene potrebno je eksplicitno definirati svaku adresu za koju poslužitelj prihvaća poštu. Definicija može biti u običnoj tekstualnoj datoteci ili bazi podataka.

Virtualne domene za prosljeđivanje koriste se da bi se elektronička pošta prosljedila s jedne adrese na drugu. Prosljeđivanje se definira kao i virtualne domene, u datoteci ili bazi podataka.

4.7.1. Podešavanje

Tablice i korisnički račun u bazi već su stvoreni, slijedi podešavanje *postfixa* za korištenje baze podataka.

Potrebno je stvoriti novu datoteku:

```
# nano /etc/postfix/mysql-virtual-alias-maps.cf
```

U njoj se nalaze podatci za pristup bazi, te upit u bazu podataka:

```
user = postfix
password = post2012
hosts = 127.0.0.1
dbname = mailfwd
query = SELECT odredisna_adresa FROM mailfwd_studenti WHERE source='%s'
```

Slijedeći korak je stvaranje takozvane *noreplay* adrese. To je adresa za koju se odbacuju sve pristigle poruke. Služi kao izvorišna adresa za automatske poruke.

U datoteku:

```
# nano /etc/aliases
```

Potrebno je dodati:

```
devnull: /dev/null
```

Zatim stvoriti novu datoteku:

```
# nano /etc/postfix/virtual
```

U koju je potrebno smjestiti sljedeći kod:

```
noreplay@domena.com devnull
```

Pri čemu je *noreplay@domena.com* adresa s koje će se slati automatske poruke.

U ovoj datoteci mogu se navesti i ostala preusmjeravanja, npr. adresa administratora, *webmastera* i slično.

Da bi *postfix* koristio ovu datoteku, potrebno ju je prevesti u binarni oblik naredbom:

```
# postmap /etc/postfix/virtual
```

Ostatak podešavanja vrši se promjenom datoteke:

```
# nano /etc/postfix/main.cf
```

Postavke za *TLS* nalaze se nakon linije koja glasi:

```
# TLS parameters
```

Te je tu potrebno izmijeniti i dodati sljedeće linije:

```
smtpd_tls_cert_file=/etc/postfix/ssl/mail.cert
smtpd_tls_key_file=/etc/postfix/ssl/mail.key
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_tls_loglevel = 2
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
smtpd_tls_security_level = may
```

Time je poslužitelj postavljen da koristi ključeve koji su napravljeni u poglavlju 4.6. Pri tome su ključevi smješteni u mapu:

```
/etc/postfix/ssl
```

Datoteka *cacert.pem* potrebna je samo ukoliko se koristi treća metoda, Vlastiti *Certificate Authority*, te se u tom slučaju nalazi u pri tome stvorenoj mapi *DemoCA*.

Ključevi stvoreni u poglavlju 4.6. enkriptirani su, međutim, *postfix* zahtjeva neenkriptirane ključeve. Stoga je potrebno pokrenuti sljedeću naredbu:

```
# openssl rsa -in mail.key -out mail.key
```

Ključno je pobrinuti se da takvim ključevima pristup ima samo *root* korisnik sustava.

Nadalje, potrebno je definirati domene za koje je poslužitelj odgovoran, tj. one domene a koje poslužitelj prihvaća elektroničku poštu:

```
mydestination =
  domena.com
  localhost.lan
  localhost
```

Uz domene za koje poslužitelj prihvaća elektroničku poštu, potrebno je navesti i lokalnu mrežu.

Na kraju datoteke potrebno je dodati:

```
virtual_alias_maps =
  mysql:/etc/postfix/mysql-virtual-alias-maps.cf
  hash:/etc/postfix/virtual
```

Čime su definirane virtualne adrese za prosljeđivanje iz baze, te *noreplay* adresa.

```
smtpd_recipient_restrictions =
  permit_mynetworks
  reject_unknown_client_hostname
  reject_unknown_sender_domain
  reject_unauth_pipelining
  reject_unauth_destination
```

Ovdje su definirana neka ograničenja za primanje pošte. Dozvoljena je sva pošta s mreža definiranih u *mynetworks*, a odbija se pošta s domena koje ne odgovaraju IP adresi pošiljalca, pošta za čiju izvorišnu adresu nije odgovoran poslužitelj, neispravne poruke, te poruke za koje poslužitelj nije zadužen.

Potrebno je još dopustiti pristup poslužitelju:

```
# nano /etc/shorewall/rules
```

Dodati sljedeće linije:

```
#mail
```

```
SMTP/ACCEPT      net          $FW
SMTPS/ACCEPT     net          $FW
Submission/ACCEPT net          $FW
```

Te ponovo pokrenuti sigurnosnu stijenu:

```
# /etc/init.d/shorewall restart
```

Crne liste

Kako je velik dio elektroničke pošte koja se razmjenjuje na Internetu neželjena pošta (engl. *spam*), s vremenom su nastale takozvane crne liste (engl. *blacklist*). To su liste IP adresa s kojih se šalje neželjena pošta. Te liste javno su dostupne i služe kao usluga poslužiteljima elektroničke pošte u cilju smanjenja prometa neželjene pošte.

Da bi *postfix* koristio takvu listu potrebno je izmijeniti postavke:

```
# nano /etc/postfix/main.cf
```

U dijelu koji se bavi ograničenjem za primanje pošte treba dodati:

```
smtpd_recipient_restrictions =
  permit_mynetworks
  reject_rbl_client dnsbl.sorbs.net
  reject_rbl_client bl.spamcop.net
  reject_rbl_client zen.spamhaus.org
  reject_rbl_client dnsbl-1.uceprotect.net
  reject_unknown_client_hostname
  reject_unknown_sender_domain
  reject_unauth_pipelining
  reject_unauth_destination
```

Svaka instanca *reject_rbl_client* provjerava jednu od crnih listi nalazi li se IP adresa s koje dolazi pošta na njoj. U ovom slučaju koriste se četiri liste, no moguće je koristiti proizvoljan broj. Treba imati na umu da se, iako rijetko, događa da se na crnim listama pojave legitimni poslužitelji, što rezultira neisporučenom poštom.

Sive liste

Još jedan način za smanjenje broja neželjenih poruka jesu sive liste (engl. *graylist*). Dok crne liste potpuno odbacuju dolaznu poštu, sive liste to čine privremeno. Nakon što stigne poruka s nove adrese, poslužitelju s koje dolazi pošta javlja se privremeno odbijanje pošte. Pravilno postavljeni poslužitelj elektroničke pošte takvu će poštu pokušati ponovo poslati nakon par minuta, međutim virusi (odnosno crvi) koji šalju neželjenu poštu često to čine samo jednom.

Nakon što je pošta s nekog poslužitelja prihvaćena jednom, sva naredna pošta koja stiže s tog poslužitelja prihvaća se odmah.

Očiti nedostatak ove metode je kašnjenje isporuke elektroničke pošte, pogotovo na novom poslužitelju koji će svu pristiglu poštu zadržavati. Na administratoru je da odluči jesu li korisnici sustava spremni prihvatiti takvu situaciju.

Postoji više načina za implementaciju sivih lista, ovdje će se koristiti paket *tumgreyspf* instaliran na početku ovog poglavlja.

Da bi *postfix* koristio *tumgreyspf* potrebno je stvoriti novi servis u datoteci:

```
# nano /etc/postfix/master.cf
```

Na kraj datoteke potrebno je dodati:


```
tumgreyspf unix - n n - - spawn
user=tumgreyspf argv=/usr/bin/tumgreyspf
```

Te u glavnu datoteku s postavkama:

```
# nano /etc/postfix/main.cf
```

Dodati:

```
smtpd_recipient_restrictions =
  permit_mynetworks
  reject_rbl_client dnsbl.sorbs.net
  reject_rbl_client bl.spamcop.net
  reject_rbl_client zen.spamhaus.org
  reject_rbl_client dnsbl-1.uceprotect.net
  check_policy_service unix:private/tumgreyspf
  reject_unknown_client_hostname
  reject_unknown_sender_domain
  reject_unauth_pipelining
  reject_unauth_destination
```

Nakon toga, potrebno je ponovo pokrenuti poslužitelj:

```
# /etc/init.d/postfix restart
```

4.7.2. Testiranje

Za testiranje sustava, dovoljno je poslati poruku na adresu studenta unesenog u testu poglavlja 4.5.

Ta poruka treba se proslijediti na određenu adresu studenta. Time je testirano primanje i slanje elektroničke pošte.

4.8. Apache2

Web stranice posluživati će *apache2* poslužitelj sa *mod_wsgi* i *mod_ssl* modulima.

4.8.1. Podešavanje

Prvi korak je uključivanje *mod_ssl* modula:

```
# a2enmod ssl
```

Preostalo podešavanje biti će izvršeno u datoteci:

```
# nano /etc/apache2/conf.d/mailfwd.conf
```

U tu datoteku potrebno je smjestiti slijedeći kod:

```
User website
Group website

WSGIDaemonProcess www.domena.com python-
path=/home/website/mailfwd:/usr/local/lib/python2.7/site-packages

<VirtualHost *:80>
  ServerName www.domena.com
  Redirect permanent / https://www.domena.com/
</VirtualHost>

<VirtualHost *:443>
  ServerName www.domena.com
  SSLEngine On
  SSLCertificateFile /etc/apache2/ssl/website.cert
  SSLCertificateKeyFile /etc/apache2/ssl/website.key
```

```

Alias /robots.txt /home/website/mailfwd/static/robots.txt
Alias /favicon.ico /home/website/mailfwd/static/favicon.ico

AliasMatch ^/([\^/]*\.css) /home/website/mailfwd/static/css/$1

Alias /media/ /home/website/mailfwd/media/
Alias /static/admin/
/usr/share/pyshared/django/contrib/admin/static/admin/
Alias /static/ /home/website/mailfwd/static/

<Directory /home/website/mailfwd/static>
    Order deny,allow
    Allow from all
</Directory>

<Directory /home/website/mailfwd/media>
    Order deny,allow
    Allow from all
</Directory>

WSGIScriptAlias / /home/website/mailfwd/mailfwd/wsgi.py

<Directory /home/website/mailfwd/mailfwd>
    <Files wsgi.py>
        Order deny,allow
        Allow from all
    </Files>
</Directory>
</VirtualHost>

```

Potrebno je još omogućiti pristup poslužitelju u sigurnosnoj stijeni:

```
# nano /etc/shorewall/rules
```

Dodati linije:

```
#web
Web/ACCEPT net $FW
```

Te ponovo pokrenuti sigurnosnu stijenu:

```
# /etc/init.d/shorewall restart
```

Nakon toga, potrebno je podesiti neke postavke u datoteci:

```
# nano /home/website/mailfwd/mailfwd/settings.py
```

Potrebno je maknuti DEBUG opciju:

```
DEBUG = False
```

Te omogućiti sljedeće opcije:

```
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
```

4.8.2. Testiranje

Za testiranje poslužitelja potrebno se spojiti na adresu poslužitelja s preglednikom Weba.

4.9. Skripte

Da bi poslužitelj dugoročno funkcionirao potrebne su neke skripte koje se brinu o tome da se ne gomilaju bespotrebni podatci.

Također je potrebna skripta za enkripciju lozinki na poslužitelju.

4.9.1. Sjednice

Svaki put kad se korisnik prijavi na poslužitelj stvori se nova sjednica. Ukoliko se korisnik odjavi klikom na odgovarajuću poveznicu, ta se sjednica briše. Međutim, ne rijetko se događa da korisnik jednostavno ugasi preglednik. U tom slučaju, sjednica ostaje u bazi podataka. Stoga je potrebna skripta da bi se takve sjednice brisale s poslužitelja.

4.9.2. Potvrda adrese elektroničke pošte

Svaki put kad korisnik promijeni adresu elektroničke pošte, na tu adresu se šalje poruka s poveznicom za potvrdu. Također, taj zahtjev zapisuje se u bazu podataka. Ukoliko korisnik ne aktivira poveznicu u roku od 12 sati, ona prestaje vrijediti. Takvi zahtjevi kojima je istekao rok valjanosti brišu se iz baze skriptom.

4.9.3. Zaboravljena lozinka

Izmjena zaboravljene lozinke vrši se tako što se korisniku šalje poruka na adresu elektroničke pošte zapisanu u bazi. U toj poruci nalazi se poveznica čijom aktivacijom korisnik može promijeniti lozinku. Kao i kod promjene adrese, i ovdje poveznica vrijedi 12 sati, te se nakon tog perioda briše iz baze ovom skriptom.

4.9.4. Enkripcija lozinki

Bcrypt

Lozinke se u bazu podataka spremaju enkriptirane. Za taj postupak koristi se *bcrypt* algoritam. Jedna od najvažnijih karakteristika tog algoritma je što se njegovo trajanje može podešavati.

Najčešće, kad se koristi enkripcija, brzina je važan faktor. Stoga su algoritmi za enkripciju uglavnom vrlo efikasni. Međutim, kad se radi o autorizaciji, brzina uglavnom nije bitna, štoviše brzi algoritam loš je izbor za enkripciju lozinki. To je stoga što ukoliko baza podataka u kojoj su zapisane lozinke korisnika padne u ruke napadača, što je algoritam brži, to je brži postupak razbijanja lozinki.

Zato je razvijen *bcrypt*. To je algoritam za enkripciju koji ima varijabilno vrijeme izvođenja. Tako je moguće stvoriti lozinke za koje treba (relativno) dugo vrijeme za dekripciju, a kako se snaga računala povećava, tako se i složenost postupka može povećati.

Nezavisna enkripcija

Kako, ovisno o računalu, enkripcija lozinki može trajati i preko sekunde po unosu, unos velikog broja studenata u sustav trajao bi vrlo dugo, pri čemu bi se najveći dio vremena trošio na samu enkripciju. Zato se lozinke u sustav unose u čitljivom tekstu, a enkriptiraju se naknadno.

Još jedna stvar koju je važno uzeti u obzir je to što je sama enkripcija procesorski vrlo zahtjevan posao, te je mnogo bolje izvršiti ga u vrijeme kada u sustavu ima malo korisnika. To je još jedan razlog da se enkripcija vrši skriptom, nezavisno od unosa.

4.9.5. Skripte

Same skripte dolaze sa *Django* aplikacijom i nalaze se u mapi:

```
/home/website/mailfwd/scripts
```

Cijeli postupak se svodi na dvije skripte:

```
bcrypt_passwords.py
database_clean.sh
```

Prva, *bcrypt_password.py*, enkriptira sve neenkriptirane lozinke u bazi, dok druga, *database_clean.sh*, briše sve zastarjele zapise iz baze.

U datoteci:

```
# nano /home/website/mailfwd/scripts/database_clean.sh
```

Potrebno je upisati podatke u slijedeća polja:

```
dbname=""
user=""
password=""
```

Pri čemu je *dbname* ime baze podataka, *user* ime korisnika baze podataka s pravima za pisanje i *password* lozinka tog korisnika.

4.9.6. cron

Cron je *Linux* proces (*daemon*) čija je zadaća pokretanje zadanih naredbi, takozvanih poslova, u određeno vrijeme. Najčešće se koristi za poslove koji se moraju redovito (dnevno, tjedno, mjesečno) pokretati na poslužiteljima.

Da bi se dodao novi posao, potrebno je pokrenuti naredbu:

```
# crontab -e
```

Otvoriti će se datoteka u programu za obradu teksta u kojoj je moguće dodavati poslove. Potrebno je dodati dva posla:

```
0 0 * * * /home/website/mailfwd/scripts/database_clean.sh
0 1 * * * /home/website/mailfwd/scripts/bcrypt_passwords.py
```

Posao u *cronu* sastoji se od pet brojeva i naredbe, brojevi označavaju:

- minutu
- sat
- dan u mjesecu
- mjesec
- dan u tjednu

Zvjezdica označava svaki (sat, dan...). Dakle, gornje postavke pokreću skripte svaki dan. U ponoć se pokreće skripta za brisanje unosa iz baze, a u jedan sat poslije ponoći pokreće se skripta za enkripciju lozinke.

5. Izvorni kod

5.1. Organizacija koda

Izvorni kod *Django* logički je podijeljen u tri skupine, modele, predloške i pogleda.

Modeli sadrže opis tablica u bazi podataka. To su klase u *pythonu* čiji članovi opisuju polja u tablici. Time je omogućeno jednostavno povezivanje izvornog koda sa sadržajem baze podataka.

Predlošci sadrže *HTML* kod Web stranica koje se poslužuju proširen sa raznim poljima koja se dinamički popunjavaju prilikom interpretacije predloška.

Pogledi sadržavaju *python* kod koji je zadužen za dinamičko generiranje sadržaja. To su metode koje prihvataju zahtjeve i vraćaju *HTML* kod koji se prikazuje u pregledniku.

5.2. *urls.py*

To je specijalna datoteka u kojoj se nalazi popis poveznica koje Web stranica koristi. U njoj se nalazi lista parova poveznica i metoda koje se pozivaju kad korisnik uputi preglednik na neku adresu.

Poveznice su prikazane kao regularni izrazi, svaki zahtjev koji se može prikazati kao dani regularni izraz upućuje se na adekvatnu metodu.

Lista se pretražuje od vrha prema dnu, stoga je potrebno specifične izraze staviti na vrh, a one generalne na dno.

5.3. Administracijski dio

Ovdje će biti opisan kod koji se bavi upravljanjem bazom podataka, dodavanjem, brisanjem i promjenom unosa.

Izvorni kod nalazi se u mapi:

```
/home/website/mailfwd/mailfwd/administration/
```

5.3.1. Modeli

U *Djangu*, model predstavlja opis tablice iz baze podataka kao klasu u *python* programskom jeziku. Modeli se nalaze u datoteci:

```
/home/website/mailfwd/mailfwd/administration/models.py
```

U ovoj aplikaciji postoje tri osnovna modela, ustanove, korisnici i studenti. Svaki student pripada nekoj ustanovi, dok se svaki korisnik brine o administraciji jedne ili više ustanova.

Osim samih polja u bazi podataka, u klasama se nalaze i neke dodatne informacije. Te se informacije nalaze u podklasi koja se naziva *Meta*. U njoj se nalazi naziv tablice u bazi, te naziv koji se koristi u aplikaciji, u jednini i množini.

Osim toga, svaki model je i klasa u *pythonu* pa samim time može imati metode kao i svaka druga klasa.

Ustanove

Ustanova je fakultet kojem student može pripadati.

Definiran je sa dva polja:

- *id* – identifikacija ustanove, njena domena (fer.hr, ffzg.hr)
- *naziv* – puni naziv ustanove

Korisnici

Korisnici su administratori baze podataka, to su odgovorne osobe koje se brinu o podacima jedne ili više ustanova u bazi.

Ova klasa sadržava samo dva polja:

- *user* – to je polje koje "pokazuje" na polje u tablici *User*, interne baze podataka *Djanga*
- *ustanove* – je polje koje povezuje korisnika s tablicom u kojoj se nalazi lista ustanova za koju je korisnik zadužen

Ovaj model proširenje je internog modela *User*, što znači da se dio informacija nalazi u internoj tablici, a ostatak u tablici *Korisnici*.

Studenti

Ovaj model sadrži podatke o studentima. Uz očekivana polja koja opisuju studenta, u modelu se nalaze dvije metode:

- *check_password* – metoda koja provjerava odgovara li lozinka predana kao argument metode onoj zapisanoj u bazi podataka
- *save* – ovo je proširenje interne metode *save*, služi tome da se označe neenkriptirane lozinke u bazi.

ChangeMailConfirmationLinks

Ovo je model koji sadrži zapis poveznice za potvrdu adrese elektroničke pošte. Prilikom promjene adrese elektroničke pošte, na novu adresu šalje se poruka za potvrdu s poveznicom čija aktivacija mijenja adresu u bazi.

Osim poveznice, u tablici se nalaze i izvorišna i odredišna adresa, te rok trajanja poveznice.

ForgotPasswordConfirmationLinks

Slično prethodnom modelu, ovdje se spremaju poveznice za za promjenu zaboravljene lozinke.

Uz poveznicu, u tablici se nalazi izvorišna adresa, te rok trajanja poveznice.

5.3.2. Django admin

Django admin je aplikacija koja dolazi s *Djangom*, omogućava brzo i jednostavno stvaranje administratorskog sučelja.

Obavlja niz zadataka koji olakšavaju razvoj Web aplikacija, autentifikaciju, jednostavan prikaz i upravljanje bazama podataka, podrška za korisnike, grupe korisnika.

5.3.3. *admin.py*

Da bi *django admin* aplikacija prepoznala modele, potrebno ih je registrirati. To se vrši u datoteci:

```
/home/website/mailfwd/mailfwd/administration/admin.py
```

Za svaki model koji je potrebno registrirati sa *django admin* aplikacijom potrebno je napraviti odgovarajuću *admin* klasu. U toj klasi definirano je na koji način se model prikazuje u aplikaciji. U ovom diplomskom radu koriste se slijedeće karakteristike:

- *list_display* – lista stupaca iz tablice koji prikazuju u listi unosa administratorskog sučelja
- *search_fields* – lista stupaca iz tablice koji se pretražuju
- *list_filter* – lista stupaca za koje se prikazuje filter u administratorskom sučelju
- *ordering* – lista stupaca po kojima se sortiraju unosi kod prikaza
- *actions* – lista akcija koje se mogu izvršiti nad unosima
- *list_display_links* – popis stupaca koji se prikazuju kao poveznica preko koje odlazi na detaljni pregled unosa
- *save_on_top* – dodaje gumbe za spremanje promjena na vrh Web stranice, ukoliko nije uključeno, gumbi se nalaze samo na dnu
- *list_max_show_all* – maksimalni broj unosa za koje se prikazuje opcija "prikaži sve"
- *list_per_page* – broj unosa koji se prikazuju na svakoj stranici

Osim toga, koriste se i slijedeće metode:

- *queryset* – ova funkcija vraća unose iz baze podataka, služi tome da se unosi filtriraju prije prikazivanja
- *formfield_for_foreignkey* – slično kao i *queryset*, no ovdje se radi o filtru za polja Web formi
- *get_urls* – vraća poveznice koje se prikazuju na vrhu Web stranice, ovdje se koristi za dodavanje vlastitih funkcionalnosti administratorskom sučelju

Tu su još dvije klase, *UserProfileInline* i *UserProfileAdmin*, koje služe tome da se ispravno prikazuje proširenje modela *User*, modelom *Korisnici*.

Na kraju datoteke, metode *unregister* i *register* odjavljuju i prijavljuju odgovarajuće modele:

- odjavljuju se modeli *User*, zato što će se nadomjestiti s proširenim modelom *UserProfileAdmin*
- odjavljuju se model *Group* jer se ne koristi
- prijavljuju se modeli *Studenti* i *Ustanove*

5.3.4. *forms.py*

U ovoj datoteci nalaze se forme koje se koriste na Web stranicama za unos podataka.

Administratorsko sučelje koristi dvije dodatne forme:

- *CSVSettingsForm* – služi za definiranje postavki prilikom uvoza i izvoza podataka iz baze
- *FileForm* – služi za odabir datoteke na lokalnom disku za uvoz podataka

5.3.5. Pogledi i akcije

Pogledi su način na koji se dodaje funkcionalnost *Django* aplikaciji. To su metode koje primaju (barem) *http* zahtjev kao argument i vraćaju Web stranicu koja se prikazuje kao rezultat. Akcije služe istoj svrsi, samo što se ne pozivaju iz *http* zahtjeva već se izvršavaju nad nizom podataka odabranih od korisnika. Za razliku od pogleda, akcije su specifična funkcionalnost *Django admin* aplikacije.

Pogledi

Pogledi se nalaze u datoteci:

```
/home/website/mailfwd/mailfwd/administration/views.py
```

import_XML(self, request)

Ovo je pogled koji obavlja zadatak unosa studenata iz datoteke. U datoteci se studenti nalaze u *XML* formatu.

Metoda prvo provjerava je li odabran gumb *Natrag*, ukoliko jest, vraća poveznicu na prethodnu stranicu.

Ukoliko je odabran gumb *Unesi*, popunjavaju se podatci iz forme u objekt *file_form*.

Zatim je definirana klasa *xml_read_errors* u koju će biti zapisane eventualne greške prilikom unosa.

Nakon toga provjerava se valjanost podataka unesenih u formu Web stranice.

Otvora se datoteka te se ona parsira i stvara se objekt *doc* u kojem je sadržaj datoteke zapisan u obliku drveta.

Slijedi prolaz po tom drvetu pri čemu se za svakog studenta stvara novi objekt naziva *stud*, i u njega se zapisuju podatci iz datoteke.

Nakon toga slijedi provjera lozinke, ukoliko ona ne postoji stvara se nova, slučajna lozinka.

Konačno, student se zapisuje u bazu.

Ukoliko dođe do greške ona se zapisuje u objekt *xml_read_errors*.

Ako je bilo greški, prikazuje se izvješće s greškama, u suprotnom preglednik se vraća na prethodnu stranicu.

Ukoliko nije odabran nijedan gumb, znači da se stranica prikazuje prvi put, pa se stoga stvori prazna forma i prikaže odgovarajući predložak.

import_CSV(self, request)

Ovo je pogled koji obavlja zadatak unosa studenata iz datoteke. U datoteci se studenti nalaze u zarezom odvojenim poljima, jedan redak po studentu.

Metoda prvo provjerava je li odabran gumb *Natrag*, ukoliko jest, vraća poveznicu na prethodnu stranicu.

Ukoliko je odabran gumb *Unesi*, popunjavaju se podatci iz formi u objekte *CSV_settings_form* i *file_form*. Zatim se provjerava valjanost tih podataka, te se podešavaju opcije unosa.

Ukoliko opcije nisu ispravne, koriste se pretpostavljene opcije.

Nakon toga, aplikacija prolazi kroz datoteku, redak po redak, provjerava valjanost podataka te ih sprema u bazu. Ukoliko podatci nisu valjani, zapisuju se greške i nastavlja.

U tom postupku, ukoliko nije definirana lozinka u datoteci sa studentima, stvara se slučajna lozinka dužine 8 znakova.

Ukoliko se prilikom unosa pojavi greška prikazuje se izvješće o greškama, inače se ispiše poruka o uspjehu i vrati na prethodnu stranicu.

Ukoliko nije odabran nijedan gumb, znači da se stranica prikazuje prvi put, pa se stoga stvore prazne forme i prikaže odgovarajući predložak.

Akcije

Akcije se nalaze u datoteci:

```
/home/website/mailfwd/mailfwd/administration/actions.py
```

intersect(list1, list2)

Ova metoda vraća elemente prve liste koji se nalaze u drugoj listi.

exclude_from(list1, list2)

Ova metoda vraća elemente prve liste koji ne postoje u drugoj listi.

export_XML_action(description, fields, exclude)

Ova metoda vraća *export_XML* akciju.

export_XML(modeladmin, request, queryset)

Ovo je generička akcija za izvoz modela u *XML* datoteku.

Ukoliko je odabran gumb *Natrag*, ne vraćamo ništa, što se interpretira kao povratak na prethodnu stranicu u *Django admin* aplikaciji.

Ukoliko je odabran gumb *Spremi*, stvori se prazna datoteka u koju će biti zapisani podatci.

U datoteku se zapiše zaglavlje *XML* datoteke.

Zatim slijedi stvaranje sadržaja datoteke u obliku stabla. Prvo se stvori korijen. Nakon toga, stvori se stablo s tim korijenom.

Nakon toga prelazi se kroz sve objekte koji se izvoze i za svaki od njih dodaje se unos u stablo.

Stablo se zapiše u datoteku i ona se šalje korisniku.

Ukoliko nije odabran nijedan gumb, prikazuje se lista unosa koji će biti izvezeni.

export_CSV_action(description, fields, exclude, header)

Ova metoda vraća *export_CSV* akciju.

export_CSV(modeladmin, request, queryset)

Ovo je generička akcija za izvoz modela u CSV datoteku.

Ukoliko je odabran gumb *Natrag*, ne vraćamo ništa, što se interpretira kao povratak na prethodnu stranicu u *Django admin* aplikaciji.

Ukoliko je odabran gumb *Spremi*, učitaju se podatci iz forme prema kojima se postavje karakteristike izvoza. Ukoliko su podatci u formi neispravni, koriste se pretpostavljene postavke.

Nakon toga podatci se zapišu u datoteku te se ona proslijedi korisniku.

Ukoliko nije odabran nijedan gumb, prikazuje se forma s postavkama izvoza te lista unosa koji će biti izvezeni.

5.3.6. Predlošci

Predlošci *Django admin* aplikacije nalaza se u mapi:

```
/home/website/mailfwd/templates/administration
```

Ovi predlošci proširuju osnovni predložak *Django admin* aplikacije, *base_site.html*. Također, svaki od tih predložaka sadrži blok *breadcrumbs*, te blok *content*.

Breadcrumbs je prikaz lokacije sličan onom kakav se koristi za prikaz datoteka i mapa na disku računala, to je niz poveznica odvojen znakom veće od.

export_CSV.html

U bloku *content* prikazuje se forma naziva *CSV_settings_form*. Forma se prikazuje kao tablica. Nakon nje prikazuju se objekti iz liste *object_list*.

Na kraju se nalaze gumbi *Spremi* i *Natrag*.

import_CSV.html

U bloku *content* prikazuju se dvije forme, *file_form*, za odabir datoteke s diska, te *CSV_settings_form* za podešavanje postavki uvoza.

Nakon toga prikazuju se gumbi za unos i povratak na prethodnu stranicu.

import_CSV_report.html

Blok *content* ovog predloška prvo prikazuje broj uspješnih unosa, ukoliko ih je bilo, a zatim greške koje su nastale pri unosu.

5.4. Korisnički dio

Slijedi opis koda koji obavlja funkciju Web stranice namijenjene pristupu studenata.

Izvorni kod nalazi se u mapi:

```
/home/website/mailfwd/mailfwd
```

5.4.1. Polja

Forme su sastavljene od polja. *Django* omogućava stvaranje vlastitih polja. Ona se obično stvaraju kad je potrebna specijalna validacija unesenih podataka.

Polja se nalaze datoteci:

```
/home/website/mailfwd/mailfwd/fields.py
```

PhoneNumberField(Field)

Ovo je polje za unos telefonskog broja. Kako fakulteti imaju studente iz inozemstva, potrebno je prihvaćati bilo koji broj iz bilo koje zemlje.

Svi međunarodni brojevi telefona regulirani su od agencije *ITU*. Ovo polje bazirano je na *ITU-T E.164*.

Stvoren je regularni izraz koji prihvaća međunarodne telefonske brojeve. Osim što provjerava odgovara li broj pravilima, dijeli ga na dva djela, prefiks i sam broj. Prefiks je međunarodni pozivni broj, dok je sve ostalo broj telefona.

Prvo je definirana greška koja se javlja ukoliko dođe do greške prilikom validacije. Nakon toga definirana je metoda *clean*.

U toj metodi prvo se poziva metoda *clean* klase iz koje je naslijeđena klasa *PhoneNumberField*. Ukoliko je polje prazno, vraća se prazni *string*.

Ukoliko se u polju nalazi neka vrijednost, prvo se uklanjaju sve zagrade, kose crte i minusi. Zatim se provjerava odgovara li vrijednost regularnom izrazu te se javlja greška ukoliko to nije slučaj.

Ukoliko broj odgovara pravilima, funkcija vraća broj kao *string*.

5.4.2. Forme

Forme se nalaze u datoteci:

```
/home/website/mailfwd/mailfwd/forms.py
```

To su klase koje se sastoje od članova koji predstavljaju polja. Polja u ovom radu koriste sljedeće karakteristike:

- *required* – definira polje koje mora biti popunjeno, ukoliko to nije slučaj korisniku se javlja greška
- *label* – naziv koji se prikazuje na Web stranici
- *widget* – vrsta samog polja koje će se prikazati na Web stranici, na primjer *PasswordInput()* prikazati će se kao polje čija će vrijednost biti maskirana prilikom unosa.
- *help_text* – pomoćni tekst polja u kojem je pojašnjeno koja vrijednost se očekuje u polju

Kao i kod polja, neke od formi imaju definiranu metodu *clean*. U ovom radu to su polja za promjenu lozinke. U toj metodi provjerava se jesu li obje unesene lozinke jednake.

LoginForm(forms.Form)

Forma za prijavu na Web stranicu. Sadržava polja za korisničko ime i lozinku.

ChangePasswordForm(forms.Form)

Forma za promjenu lozinke. Zahtjeva unos stare lozinke, nove lozinke i potvrdu nove lozinke.

ChangeAddressForm(forms.Form)

Forma za promjenu adrese. Sadržava polja za unos adrese, poštanskog broja, mjesta i države.

ChangePhoneForm(forms.Form)

Forma za promjenu telefonskih brojeva. Sadržava polja za unos telefonskog broja iz prethodnog poglavlja.

ChangeMailForm(forms.Form)

Forma za promjenu adrese elektroničke pošte. Sadržava polje za unos adrese elektroničke pošte.

ForgotPasswordRequestForm(forms.Form)

Forma za zahtjev za promjenu zaboravljene lozinke. Sadržava polja za unos izvorišne i odredišne adrese studenta.

ForgotPasswordChangeForm(forms.Form)

Forma za promjenu zaboravljene lozinke. Sadržava polja za unos izvorišne adrese studenta, nove lozinke, potvrde nove lozinke te skriveno polje *uid*, koje se automatski popunjava prilikom prikaza forme.

5.4.3. Pogledi

Pogledi se nalaze u datoteci:

```
/home/website/mailfwd/mailfwd/views.py
```

login_user(request)

Ovo je pogled koji se studentu prikazuje kad se spoji na Web stranicu i nije prijavljen. Sadržava formu za unos korisničkog imena i lozinke.

Ukoliko je bio odabran gumb za zaboravljenu lozinku, preglednik se prousmjerava na odgovarajuću adresu.

Ukoliko je korisnik već prijavljen u sustav (postoji *member_id* u sjednici), preglednik se prousmjerava na početnu stranicu.

Stvori se prazna forma za prijavu, te se definiraju varijable za korisničko ime i lozinku.

Ukoliko je student odabrao gumb *Login*, forma se popunjava s podacima studenta, ukoliko su podatci valjani, provjerava se postoji li student s danim korisničkim imenom.

Nakon što je pronađen odgovarajući student, poziva se metoda *check_password* koja provjerava je li lozinka ispravna. Ukoliko postoji student i lozinka je ispravna, u sjednicu se zapisuje izvorišna adresa studenta, i time je završena prijava u sustav.

Ukoliko nije bio odabran nijedan gumb, ili nije uspjela neka od provjera, prikazuje se predložak *auth.html*.

logout_user(request)

Ovo je pogled koji služi odjavi korisnika s Web stranice.

Pokušava se obrisati *member_id* varijabla iz sjednice, te obrisati samu sjednicu.

Ukoliko dođe do greške ona se ignorira.

Nakon toga, preglednik se preusmjerava na prethodnu stranicu.

home(request)

Ovo je pogled koji prikazuje početnu Web stranicu studenta koji je prijavljen. Ukoliko student nije prijavljen, preusmjerava se na Web stranicu za prijavu.

U varijablu student zapisuje se objekt s podacima studenta prijavljenog u sustav.

Pošanski broj u bazu podataka zapisuje se kao cijeli broj. To znači da ukoliko poštanski broj nije naveden on je predstavljen s brojem nula.

Provjerava se ako nema poštanskog broja, te se njegova vrijednost u tom slučaju pretvara u prazni *string*.

Nakon toga, prikazuje se predložak *student.html*.

change_password(request)

Ovo je pogled koji služi promjeni lozinke studenta.

Ukoliko student nije prijavljen, preusmjerava se na Web stranicu za prijavu.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

Stvara se forma za promjenu lozinke.

Ako je odabran gumb *Promijeni*, forma se popunjava s podacima s Web stranice.

Ukoliko su podatci valjani, u varijablu *student* zapisuje se objekt s podacima studenta prijavljenog u sustav.

Provjerava se valjanost stare lozinke, te ukoliko je ona ispravna, slijedi provjera istovjetnosti nove lozinke i potvrde nove lozinke, te njen zapis u bazu. Nakon toga preglednik se preusmjerava na početnu stranicu.

Ukoliko nije bio odabran nijedan gumb ili neka od provjeri nije bila uspješna, prikazuje se predložak *change_form.html*.

change_address(request)

Ovo je pogled koji služi promjeni adrese studenta.

Ukoliko student nije prijavljen, preusmjerava se na Web stranicu za prijavu.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

U varijablu *student* zapisuje se objekt s podacima studenta prijavljenog u sustav.

Stvara se forma za promjenu adrese popunjena s podacima iz baze podataka.

Ukoliko je odabran gumb *Promijeni*, forma se popunjava s podacima s Web stranice. Ako je forma valjana, vrijednosti se zapisuju u bazu podataka.

Ukoliko nije bio odabran nijedan gumb ili neka od provjeri nije bila uspješna, prikazuje se predložak *change_form.html*.

change_phone(request)

Ovo je pogled koji služi promjeni telefonskih brojeva studenta.

Ukoliko student nije prijavljen, preusmjerava se na Web stranicu za prijavu.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

U varijablu *student* zapisuje se objekt s podacima studenta prijavljenog u sustav.

Stvara se forma za promjenu brojeva telefona popunjena s podacima iz baze podataka.

Ukoliko je odabran gumb *Promijeni*, forma se popunjava s podacima s Web stranice. Ako je forma valjana, vrijednosti se zapisuju u bazu podataka.

Ukoliko nije bio odabran nijedan gumb ili neka od provjeri nije bila uspješna, prikazuje se predložak *change_form.html*.

change_mail(request)

Ovo je pogled koji služi promjeni telefonskih brojeva studenta.

Ukoliko student nije prijavljen, preusmjerava se na Web stranicu za prijavu.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

U varijablu *student* zapisuje se objekt s podacima studenta prijavljenog u sustav.

Stvara se forma za promjenu brojeva telefona popunjena s podacima iz baze podataka.

Ukoliko je odabran gumb *Promijeni*, forma se popunjava s podacima s Web stranice.

Ako je forma valjana, provjerava se jeli adresa jednaka onoj koja se već nalazi u bazi podatka, te se u tom slučaju preglednik preusmjerava na početnu stranicu.

Ukoliko je nova adresa različita od one u bazi podataka, stvara se novi identifikator. To je slučajni niz od 96 bita enkodiran *base64* kodom, što rezultira sa 128 znakova. U nizu se znak jednako mijenja sa znamenkom nula.

Nakon toga stvara se jedinstvena poveznica za potvrdu adrese elektroničke pošte.

Stvara se nova poruka iz predloška *change_mail_confirmation_mail.html* datoteke. Taj se predložak popunjava s podacima o studentu, poveznicom i kontakt adresom.

Određuje se vrijeme do kojeg valja poveznica, tako što se trenutnom vremenu dodaje 12 sati.

Stvara se objekt modela *ChangeMailConfirmationLinks* u koji se zapisuju poveznica, izvorišna i odredišna adresa studenta te rok valjanosti poveznice, te se taj objekt spremi u bazu podataka.

Nakon toga, studentu se šalje se elektronička pošta s porukom na novu adresu.

Prikazuje se informacija o poslanoj poruci.

Ukoliko je student obrisao adresu iz forme, ona se briše iz baze podataka i preglednik se preusmjerava na početnu stranicu.

Ukoliko nije bio odabran nijedan gumb ili neka od provjeri nije bila uspješna, prikazuje se predložak *change_form.html*.

change_mail_confirm(request, uid)

Ovo je pogled koji služi potvrdi promjene adrese elektroničke pošte studenta. Za razliku od ostalih pogleda, ovaj pogled prihvaća jednu dodatnu varijablu, *uid*.

Pokušava se stvoriti objekt s podacima iz baze podataka čiji je uid jednak onom iz poziva pogleda.

Ukoliko nema takvog zapisa u bazi, korisniku se dojavljuje greška 404, stranica nije pronađena. Do ove situacije može se doći jedino upisivanjem nevažeće adrese za potvrdu promjene adrese elektroničke pošte.

Ukoliko zapis u bazi postoji, provjerava se njegova valjanost. Ukoliko je rok valjanosti istekao, zapise se briše i korisniku se javlja greška 404.

Ako zapis postoji i valjan je, u varijablu *student* zapisuje se objekt s podacima studenta prijavljenog u sustav, mijenja se adresa tog objekta i on se zapisuje u bazu, čime je potamnjena adresa elektroničke pošte studenta.

Poveznica za promjenu se briše iz baze podataka i preglednik se preusmjerava na početnu stranicu.

forgot_password(request)

Ovo je pogled koji služi promjeni zaboravljene lozinke studenta.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

Stvara se forma za izmjenu zaboravljene lozinke.

Ako je odabran gumb *Promijeni*, forma se popunjava s podacima s Web stranice.

Ukoliko je forma ispravna te postoje izvorišna i odredišna adresa, u varijablu *student* pokušava se zapisati objekt s podacima studenta sa zadanom izvorišnom adresom.

Ako takav student postoji, provjerava se odgovara li odredišna adresa tog studenta onoj upisanoj u formu. Ukoliko su adrese iste, stvara se novi identifikator. To je slučajni niz od 96 bita enkodiran base64 kodom, što rezultira sa 128 znakova. U nizu se znak jednako mijenja sa znamenkom nula.

Stvara se jedinstvena poveznica za promjenu zaboravljene lozinke.

Nakon toga, stvara se nova poruka iz predloška *forgot_password_confirmation_mail.html* datoteke, te se ona popunjava s imenom studenta, poveznicom i kontakt adresom.

Određuje se vrijeme do kojeg valja poveznica, tako što se trenutnom vremenu dodaje 12

sati.

Stvara se objekt modela *ForgotPasswordConfirmationLinks* u koji se zapisuju poveznica, izvorišna i rok valjanosti poveznice, te se taj objekt spremi u bazu podataka.

Nakon toga, studentu se šalje se elektronička pošta s porukom.

Prikazuje se informacija o poslanoj poruci.

Ukoliko nije bio odabran nijedan gumb ili neka od provjeri nije bila uspješna, prikazuje se predložak *change_form.html*.

forgot_password_confirm(request, uid)

Ovo je pogled koji služi promjeni zaboravljene lozinke studenta. Za razliku od ostalih pogleda, ovaj pogled prihvaća jednu dodatnu varijablu, *uid*.

Ukoliko je odabran gumb *Natrag*, preglednik se vraća na prethodnu stranicu.

Stvara se forma sa varijablom *uid* zapisanom u skriveno polje istog naziva.

Ako je odabran gumb *Promijeni*, forma se popunjava s vrijednostima s Web stranice.

Ukoliko je forma ispravna, provjerava se jesu li sva polja popunjena, te se pokušava stvoriti objekt modela *ForgotPasswordConfirmationLinks* čiji *uid* je jednak onom s Web stranice. Ukoliko nema takvog zapisa u bazi prikazuje se greška 404.

Ukoliko je poveznici istekao rok trajanja, prikazuje se greška 404 i zapis se briše iz baze podataka.

U varijablu *student* zapisuje se objekt s podacima studenta čija je izvorišna adresa upisana u formu. Provjerava se odgovara li izvorišna adresa tog studenta adresi studenta koji je zatražio promjenu lozinke. Ova provjera služi tome da jedan korisnik ne bi bio u mogućnosti zatražiti promjenu lozinke sa svojim podacima, a zatim kod promjene napisati ispravnu adresu drugog studenta i promijeniti njegovu lozinku.

Vrši se usporedba nove lozinke i potvrde nove lozinke, te se ista zapisuje u bazu ukoliko su jednake.

Briše se poveznica za promjenu zaboravljene lozinke iz baze podataka, te se preglednik vraća na početnu stranicu.

Ukoliko forma nije bila potpuno popunjena, prikazuje se greška 400, krivi zahtjev. To je moguće samo ukoliko se ručno mijenjaju podatci koji se šalju na poslužitelj.

Ako nije odabran nijedan gumb, provjerava se postoji li poveznica u bazi podataka, te se prikazuje greška 404 ukoliko ne postoji.

Zatim se provjerava je li istekao rok trajanja poveznici, te se u tom slučaju poveznica briše iz baze podataka i prikazuje se greška 404.

Ukoliko poveznica postoji i valjana je, prikazuje se predložak *change_form.html*.

5.4.4. Predlošci

Predlošci se nalaze u mapi:

```
/home/website/mailfwd/templates
```


base.html

Ovo je osnovni predložak koji se nasljeđuje u ostalim predlošcima. U njemu se nalazi glava Web stranice u kojoj je blok *title*. Tu je i tijelo Web stranice s blokovima *header* i *content*.

Blok *title* je naslov Web stranice.

Blok *header* je zaglavlje Web stranice.

Blok *content* je sadržaj Web stranice.

auth.html

Ovaj predložak koristi se za prijavu studenata u sustav.

Predložak proširuje blok *content* osnovnog predloška, *base.html*.

U sadržaju Web stranice, prvo se prikazuje varijabla *state*. Tu se prikazuju greške prilikom prijave.

Zatim se prikazuje forma kao tablica. Za svako polje prikazuje se jedan redak u tablici.

Na kraju se nalaze gumbi za prijavu i promjenu zaboravljene lozinke.

student.html

Ovaj predložak koristi se za početnu stranicu nakon prijave u sustav.

Predložak proširuje blokove *header* i *content* osnovnog predloška, *base.html*.

U bloku *header* prikazuje se poveznica za odjavu iz sustava.

U bloku *content* prikazuju se podatci o studentu, ime, prezime, adresa elektroničke pošte, ustanova te godina diplomiranja.

Zatim se prikazuje tablica s podacima koje student može mijenjati. Svaki redak sastoji se od naziva, vrijednosti, tj. samog podatka, te poveznice za promjenu tog podatka u obliku ikone.

change_form.html

Ovaj predložak koristi se za promjenu podataka studenta.

Predložak proširuje blokove *header* i *content* osnovnog predloška, *base.html*.

U bloku *header* prikazuje se poveznica za odjavu iz sustava, ukoliko je student prijavljen u sustav.

U bloku *content* prikazuje se greška ukoliko one postoje, te se zatim prikazuje forma kao tablica. Za svako polje forme prikazuje se jedan redak tablice.

Nakon tablice prikazuju se gumbi *Promijeni* i *Natrag*.

confirmation_sent.html

Ovaj predložak koristi se za izvješće o poslanoj poruci elektroničke pošte s poveznicom koju je potrebno slijediti za nastavak.

Predložak proširuje blokove *header* i *content* osnovnog predloška, *base.html*.

U bloku *header* prikazuje se poveznica za odjavu iz sustava.

U bloku *content* prikazuje se poruka korisniku i gumb za povratak.

change_mail_confirmation_mail.html,
forgot_password_confirmation_mail.html

Ovi predložak koristi se za poruke elektroničke pošte u kojoj se šalju poveznice za potvrdu određene adrese studenta i promjenu zaboravljene lozinke.

U njoj se nalaze varijable za ime studenta, određenu adresu, poveznicu za promjenu te kontakt adresu.

400.html, *404.html*, *500.html*

Ovo su predlošci koji se koriste za greške poslužitelja.

Predlošci proširuju blokove *title* i *content* osnovnog predloška, *base.html*.

U bloku *title* nalazi se naslov, ime greške.

U bloku *content* nalazi se kratko pojašnjenje greške.

5.5. CSS

CSS je način za dizajn i oblikovanje izgleda Web stranice. CSS datoteke nalaze se u mapi:

`/home/website/mailfwd/static/css/`

Ovaj diplomski rad neće se baviti dizajnom. CSS kod koji dolazi sa aplikacijom služi samo da poboljša dojam, dok će kod primjene vjerojatno biti prilagođen dizajnu sustava u sklopu kojeg će djelovati.

6. Korištenje sustava

6.1. Super korisnik i korisnici

Administracijsko sučelje predviđeno je da bi ga koristile osobe od povjerenja. Moguće je imati više korisnika koji mogu imati različite razine pristupa različitim tablicama u bazi podataka.

6.1.1. Spajanje na sustav

Korisnici i super korisnici na sustav se spajaju usmjeravanjem preglednika na adresu:

`www.domena.com/admin`

6.1.2. Super korisnik

Super korisnik je osoba koja ima apsolutna prava na izmjenu podataka u bazi. Taj korisnik ima pravo dodavati, mijenjati i brisati bilo koji podatak u bilo kojoj tablici.

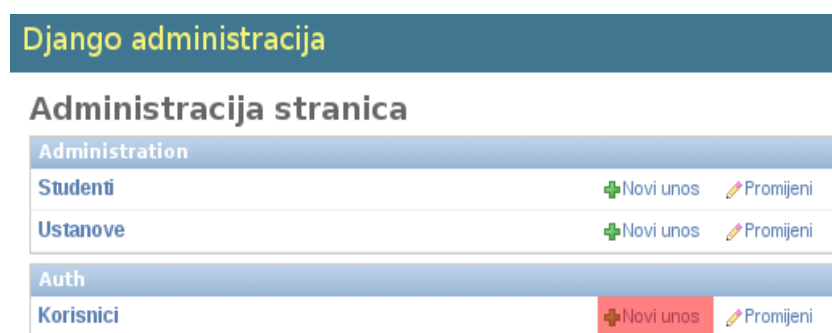
Također, super korisnik može dodavati ostale korisnike (pa i druge super korisnike), određivati njihova prava, i slično.

6.1.3. Korisnici

Korisnik je osoba od povjerenja koja ima pristup nekoj od tablica ustanova (ili više njih) u bazi podataka. To su osobe zadužene za dodavanje i ažuriranje podataka koji se prikazuju na web stranici.

Dodavanje korisnika

Za dodavanje korisnika potrebno se spojiti na administracijsko sučelje kao super korisnik. Zatim kliknuti na *Novi unos*:



Pojaviti će se forma u koju je potrebno unijeti korisničko ime i lozinku novog korisnika.

Django administracija

Početna > Auth > Korisnici > Novi unos korisnik

Novi unos (korisnik)

Prvo, unesite korisničko ime i lozinku. Onda možete promijeniti više postavki korisnika.

Korisničko ime:	<input type="text"/>
	<small>Obavezan unos. 30 alfanumeričkih znakova ili manje. Samo slova, brojevi i @/./+/-/_ znakovi.</small>
Lozinka:	<input type="password"/>
Potvrda lozinke:	<input type="password"/>
	<small>Unesite istu lozinku, za potvrdu.</small>

Ispod toga nalazi se forma za izbor ustanova za koje je novi korisnik zadužen. U listi odabrati one ustanove za koje će korisnik biti zadužen te kliknuti strelicu desno.

Korisnici

Korisnik: #1

Držite "Control", ili "Command" na Mac-u, da bi odabrali više od jednog objekta.

Ustanove:

Dostupno ustanove ☺

Filter

- adu.unizg.hr
- agr.unizg.hr
- alu.unizg.hr
- alumni.fer.hr
- arhitekt.unizg.hr
- efzg.unizg.hr
- erf.unizg.hr
- fer.unizg.hr
- ffzg.unizg.hr
- fkit.unizg.hr
- foi.unizg.hr
- fpz.unizg.hr
- fpzg.unizg.hr
- fsb.unizg.hr
- geof.unizg.hr

Odaberi sve ▶

Odabrano ustanove ☺ +

Ukloni sve ◀

Nakon pritiska na tipku *Spremi* pojaviti će se nova forma na kojoj je moguće unijeti više podataka o korisniku. Tu je bitno odabrati *Status osoblja*.

Privilegije

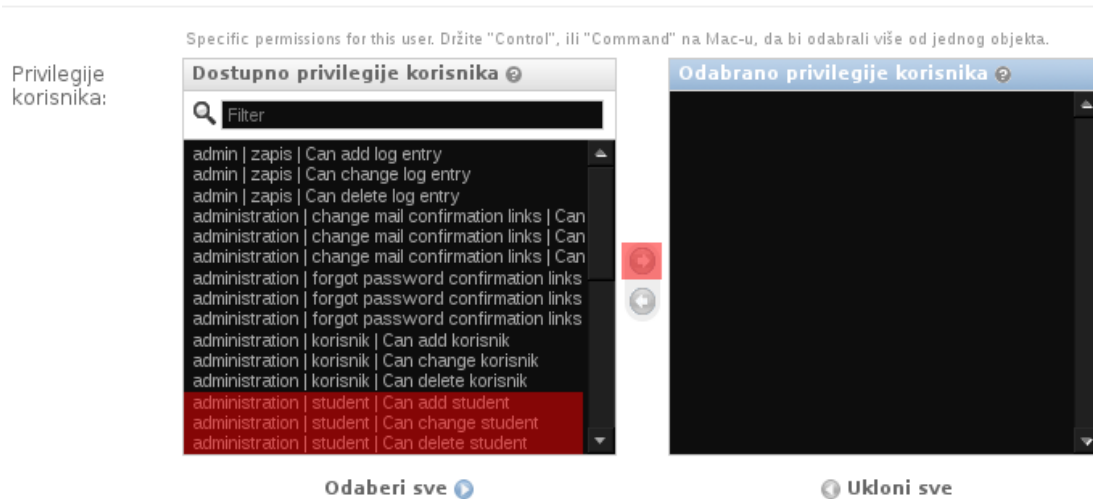
Aktivan
Određuje treba li se ovaj korisnik tretirati kao aktivan korisnik. Koristite ovu

Status osoblja
Određuje može li se korisnik prijaviti na ove stranice administracije.

Superuser status
Određuje da ovaj korisnik ima sve privilegije te uklanja potrebu da se privi

Ispod toga nalazi se lista s mogućim privilegijama korisnika. Potrebno je odabrati za koje

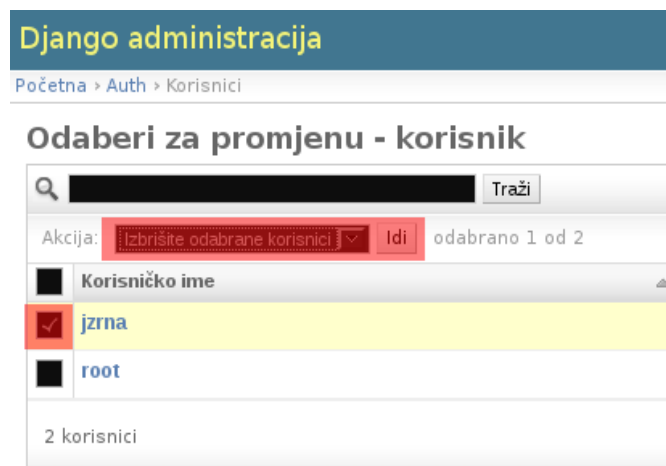
tablice korisnik ima privilegije. Odabrati sve privilegije nad tablicom *student* te pritisnuti strelicu desno. Moguće je dati i prava nad ostalim tablicama, no potrebno je imati na umu da aplikacija nije predviđena za to.



Nakon što su dodijeljene privilegije, potrebno je odabrati *Spremi*.

Brisanje korisnika

Za brisanje korisnika, potrebno je u listi korisnika odabrati one korisnike koje je potrebno izbrisati.



Zatim u izborniku akcija odabrati "Izbrišite odabrane korisnike", te kliknuti "Idi". Pojaviti će se lista s korisnicima koji će se izbrisati i gumbom za potvrdu i povratak.

Uređivanje korisnika

Za uređivanje korisnika potrebno je u listi korisnika kliknuti na ime onog korisnika čiji zapis je potrebno urediti. Nakon toga pojaviti će se forma s detaljima korisnika. Kad je

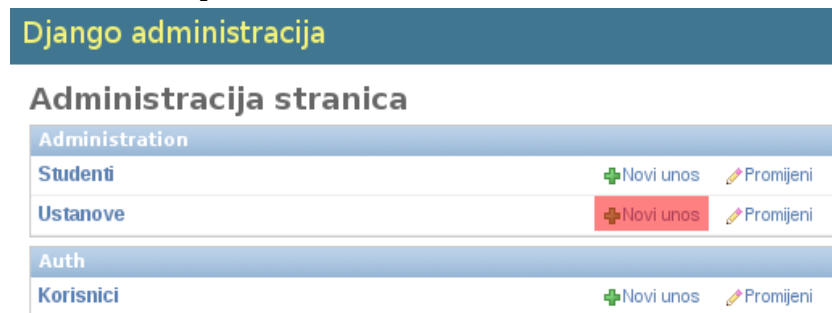
završena izmjena potrebno je pritisnuti gumb *Spremi*.

6.1.4. Ustanove

Svaki student pripada nekoj ustanovi, dok svaki korisnik ima prava nad nekom od ustanova. Ustanove dakle određuju kojim studentima korisnik ima pravo pregledavati i mijenjati podatke.

Dodavanje ustanove

Dodavanje ustanove vrši se pritiskom na *Novi unos*:



Nakon čega je potrebno navesti identifikaciju i naziv ustanove.

The screenshot shows the Django administration interface for adding a new institution. At the top, there is a blue header with the text 'Django administracija'. Below this is a breadcrumb trail: 'Početna > Administration > Ustanove > Novi unos ustanova'. The main section is titled 'Novi unos (ustanova)'. It contains two input fields: 'ID:' and 'Naziv:'. Both fields are currently empty and have a black background. Below these fields is a large, empty white box.

Kad su uneseni podaci potrebno je odabrati *Spremi*.

Brisanje ustanove

Za brisanje ustanove, u popisu ustanova potrebno je odabrati one ustanove koje je potrebno izbrisati.

Django administracija

Početna > Administration > Ustanove

Odaberi za promjenu - ustanova

🔍 Traži

Akcija: Izbrišite odabrane ustanove Idi odabrano 2 od 34

<input type="checkbox"/>	ID
<input type="checkbox"/>	vef.unizg.hr
<input checked="" type="checkbox"/>	ufzg.unizg.hr
<input type="checkbox"/>	ttf.unizg.hr
<input checked="" type="checkbox"/>	sumfak.unizg.hr
<input type="checkbox"/>	simet.unizg.hr
<input type="checkbox"/>	sfza.unizg.hr

Te zatim u akcijama odabrati "Izbrišite odabrane ustanove" i odabrati "Idi". Nakon toga prikazati će se lista s ustanovama koje su odabrane za brisanje i gumbom za potvrdu i povratak.

Uređivanje ustanove

Za uređivanje ustanova potrebno je u listi ustanova kliknuti na ime one ustanove čiji zapis je potrebno urediti. Nakon toga pojaviti će se forma s detaljima ustanove. Po završetku izmjene potrebno je pritisnuti gumb *Spremi*.

6.1.5. Studenti

Studenti su krajnji korisnici sustava. Sve usluge sustava namijenjene su upravo njima. Svaki student pripada nekoj ustanovi.

Dodavanje studenta

Dodavanje studenta vrši se pritiskom na *Novi unos*:

Django administracija

Administracija stranica

Administration	
Studenti	+ Novi unos ✎ Promijeni
Ustanove	+ Novi unos ✎ Promijeni
Auth	
Korisnici	+ Novi unos ✎ Promijeni

Nakon toga prikazati će se forma u koju je potrebno unijeti podatke o studentu. Polja čija imena su zadebljana neophodno je unijeti.

Django administracija

Početna > Administration > Studenti > Novi unos student

Novi unos (student)

<input type="text"/>	
Ime:	<input type="text"/>
Prezime:	<input type="text"/>
JMBAG:	<input type="text"/>
Ustanova:	<input type="text"/> <input type="button" value="v"/> <input type="button" value="+"/>
Godina diplomiranja:	<input type="text"/>
Adresa:	<input type="text"/>
Poštanski broj:	<input type="text"/>
Mjesto:	<input type="text"/>
Država:	<input type="text"/>
Broj telefona:	<input type="text"/>
Broj mobitela:	<input type="text"/>
Izvorišna adresa:	<input type="text"/>
Odredišna adresa:	<input type="text"/>
Lozinka:	<input type="text"/>
<input type="text"/>	

Kad su podaci uneseni potrebno je pritisnuti *Spremi*.

Brisanje studenata

Za brisanje studenta, u popisu studenata potrebno je odabrati one studente koje je potrebno izbrisati.

Django administracija

Početna > Administration > Studenti

Odaberi za promjenu - student

Akcija: **Izbrišite odabrane studente** odabrano 2 od 31

<input type="checkbox"/>	JMBAG	Ime	3 ▲	Prezime
<input type="checkbox"/>	6389333450	Jasnica		Bencetić
<input checked="" type="checkbox"/>	5030495732	Marijana		Dobrić
<input type="checkbox"/>	0036417611	Jurica		Zrna
<input checked="" type="checkbox"/>	0684365522	Alan		Hajdinjak
<input type="checkbox"/>	7316219728	Tina		Kutleša
<input type="checkbox"/>	4023815696	Nerina		Mikić
<input type="checkbox"/>	2487009515	Ariana		Žanar

Te zatim u akcijama odabrati "Izbrišite odabrane studente" i odabrati "Idi". Nakon toga prikazati će se lista sa studentima koji su odabrani za brisanje i gumbima za potvrdu i povratak.

Uređivanje studenata

Za uređivanje studenta potrebno je u listi studenata kliknuti na ime onog studenta čiji zapis je potrebno urediti. Nakon toga pojaviti će se forma s detaljima studenta. Po završetku izmjene potrebno je pritisnuti gumb *Spremi*.

Uvoz studenata

Unašanje svih studenata nekog fakulteta bilo bi iznimno zahtjevan posao ukoliko bi se vršilo preko Web forme. Kako su svi fakulteti u današnje vrijeme informatizirani, informacije o svim studentima već se nalaze u barem jednoj bazi podataka. Da bi se izbjeglo ručno prepisivanje velikih količina podataka koji su već digitalizirani, aplikacija je u mogućnosti prihvatiti veliki broj studenata u dva prihvaćena i rasprostranjena formata.

Prvi je *CSV* (engl. *comma separated value*), odnosno zarezom odvojene vrijednosti. To je datoteka u kojoj je svaki student prikazan jednim retkom, u kojem su vrijednosti pojedinih informacija o studentu odvojene zarezom, tj. nekim znakom.

Iako je u nazivu formata definiran zarez, vrijednosti mogu biti odvojene bilo kojim znakom. Također, vrijednosti mogu biti zatvorene u navodnike. Vrijednosti tih znakova mogu se podesiti prilikom uvoza.

Drugi format je *XML* (engl. *extensible markup language*). To je format u kojem su vrijednosti zapisane u obliku "stabla". Format je u širokoj upotrebi i mnogi programi podržavaju uvoz i izvoz podataka u tom formatu.

Izvoz studenata

Kako bi podatci u sustavu bili dostupni i drugim sustavima, moguće je napraviti izvoz studenata. Za izvoz su dostupni isti formati kao i za uvoz.

Uvoz studenata iz CSV datoteke

Za uvoz studenata iz CSV datoteke potrebno je odabrati "Uvoz CSV":



Nakon toga prikazati će se forma za odabir datoteke s podacima, te postavki uvoza.

 The screenshot shows the 'Uvoz iz CSV datoteke' form. The title is 'Uvoz iz CSV datoteke'. Under the heading 'Opcije:', there are four input fields: 'Datoteka:' with a 'Browse...' button, 'Zaglavlje:', 'Polja završena sa:', and 'Polja zatvorena u:'. At the bottom, there are two buttons: 'Unesi...' and 'Natrag'.

Potrebno je odabrati datoteku i postaviti postavke s obzirom na datoteku te odabrati *Unesi*. Ukoliko uvoz prođe bez greške preglednik će biti vraćen na prethodnu stranicu, u suprotnom prikazati će se izvješće s greškama pri uvozu.

Uvoz studenata iz XML datoteke

Za uvoz studenata iz XML datoteke potrebno je odabrati "Uvoz XML":



Nakon toga prikazati će se forma za odabir datoteke s podacima.

Django administracija

Početna > Administration > Studenti > Import from XML

Uvoz iz XML datoteke

Datoteka:

Potrebno je odabrati datoteku te odabrati *Unesi*. Ukoliko uvoz prođe bez greške preglednik će biti vraćen na prethodnu stranicu, u suprotnom prikazati će se izvješće s greškama pri uvozu.

Izvoz studenata u CSV datoteku

Za izvoz studenata, potrebno je označiti studente koje je potrebno izvesti u popisu studenata, te zatim u akcijama odabrati "Spremi označene studente kao CSV":

Django administracija

Početna > Administration > Studenti

Odaberi za promjenu - student

Akcija: odabrano 3 od 31

<input type="checkbox"/>	JMBAG	Ime	3 ▲	Prezime	2 ▲	Godina dij
<input checked="" type="checkbox"/>	6389333450	Jasnica		Bencetić		2002
<input type="checkbox"/>	5030495732	Marijana		Dobrić		1986
<input checked="" type="checkbox"/>	0036417611	Jurica		Zrna		2012
<input type="checkbox"/>	0684365522	Alan		Hajdinjak		2012
<input checked="" type="checkbox"/>	7316219728	Tina		Kuteša		1991
<input type="checkbox"/>	1023815696	Norina		Mikić		2002

Prikazati će se forma s postavkama za izvoz i popisom studenata koji su označeni za izvoz:

Django administracija Dobrodo

Početna > Administration > Studenti > Export to CSV

Izvoz u CSV

Opcije:

Zaglavlje:

Polja završena sa:

Polja zatvorena u:

Biti će spremljeni sljedeći studenti:

- 6389333450 Bencetić, Jasnica - adu.unizg.hr - jasnica.bencetic@hornwood.dyndns.org
- 0036417611 Zrna, Jurica - alumni.fer.hr - jurica.zrna@hornwood.dyndns.org
- 7316219728 Kutleša, Tina - efzg.unizg.hr - tina.kutlesa@hornwood.dyndns.org

Potrebno je postaviti željene postavke i odabrati *Spremi*.

Izvoz studenata u XML datoteku

Za izvoz studenata, potrebno je označiti studente koje je potrebno izvesti u popisu studenata, te zatim u akcijama odabrati "Spremi označene studente kao XML":

Django administracija

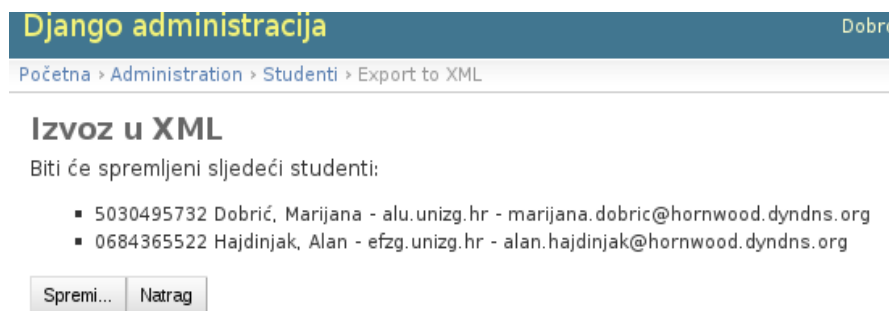
Početna > Administration > Studenti

Odaberi za promjenu - student

Akcija: Spremi označene studente kao XML odabrano 2 od 31

<input type="checkbox"/>	JMBAG	Ime	Prezime	God
<input type="checkbox"/>	6389333450	Jasnica	Bencetić	2002
<input checked="" type="checkbox"/>	5030495732	Marijana	Dobrić	1986
<input type="checkbox"/>	0036417611	Jurica	Zrna	2012
<input checked="" type="checkbox"/>	0684365522	Alan	Hajdinjak	2012
<input type="checkbox"/>	7316219728	Tina	Kutleša	1991

Prikazati će se lista s popisom studenata koji su označeni za izvoz:



Django administracija Dobrić

Početna > Administration > Studenti > Export to XML

Izvoz u XML

Biti će spremljeni sljedeći studenti:

- 5030495732 Dobrić, Marijana - alu.unizg.hr - marijana.dobric@hornwood.dyndns.org
- 0684365522 Hajdinjak, Alan - efzg.unizg.hr - alan.hajdinjak@hornwood.dyndns.org

Spremi... Natrag

Potrebno je odabrati *Spremi*.

6.2. Studenti

Studenti se na sustav spajaju kako bi promijenili svoje podatke.

6.2.1. Spajanje na sustav.

Studenti se na sustav spajaju usmjeravanjem preglednika na adresu:

6.2.2. Promjena podataka

Slijedi opis postupka promijene podataka. Za promjenu bilo kojeg podatka treba odabrati gumb sa slikom zupčanika koji se nalazi uz podatak koji je potrebno promijeniti.



Jurica Zrna

jurica.zrna@alumni-test.zemris.fer.hr

Fakultet elektrotehnike i računarstva - 2012.

e-mail: diplomski.forward.test@gmail.com	
Adresa: Unska 3	
Poštanski broj: 10000	
Mjesto: Zagreb	
Država: Hrvatska	
Telefon:	
Mobitel:	
Lozinka: *****	

Nakon toga prikazuje se forma za promjenu podataka.

Promjena odredišta prosljeđivanja

Potrebno je upisati adresu elektroničke pošte i odabrati *Promjeni*. Nakon toga pojaviti će se obavijest da je na novu adresu poslana poruka. Potrebno je otvoriti elektroničku poštu na odredišnoj adresi i slijediti poveznicu u toj poruci.

Aktivacijom poveznice iz poruke adresa će biti izmijenjena.

Promjena adrese i kontakt telefona

Potrebno je upisati podatke u polja na Web stranici te odabrati *Promjeni*.

Promjena lozinke

Za promjenu lozinke potrebno je upisati staru lozinku, te zatim novu lozinku dva puta. Nakon toga odabrati *Promijeni* da bi se nova lozinka spremila u sustav.

6.2.3. Zaboravljena lozinka

Ukoliko student zaboravi lozinku, nakon spajanja na Web stanicu sustava treba odabrati

"Zaboravljena lozinka".



Korisničko ime:

Lozinka:

[Log In](#) [Zaboravljena lozinka](#)

Nakon toga pojaviti će se forma za unos adrese elektroničke pošte sustava i adrese elektroničke pošte na koju se prosleđuju poruke.



Studentska e-mail adresa:

Odredišna e-mail adresa:

[Promjeni](#) [Natrag](#)

Nakon unosa adresa potrebno je odabrati *Promijeni*, što rezultira porukom s poveznicom za promjenu lozinke poslanom na odredišnu adresu studenta. Nakon što student slijedi poveznicu iz poruke prikazuje se forma za promjenu lozinke.



Studentska e-mail adresa:

Nova lozinka:

Nova lozinka

Nova lozinka:

Ponovite novu lozinku

Potrebno je upisati adresu elektroničke pošte studenta, te dva puta lozinku i odabrati *Promijeni*, nakon čega će student biti u mogućnosti prijaviti se u sustav s novom lozinkom.

7. Zaključak

U ovom radu prikazan je postupak izgradnje sustava za upravljanje elektroničkom poštom. Sustav je postavljen tako da korisnici i administratori sustava njime mogu upravljati preko Web sučelja.

U obzir je uzeta činjenica da nepažljivi korisnici lako mogu napraviti štetu u sustavu, te je u svakom koraku vođena briga o sigurnosti.

Besplatni alati omogućavaju jeftinu izgradnju sustava, a njihov kvalitetan izbor osigurava jednostavno održavanje. Otvoreni kod osigurava prilagodljivost sustava novim potrebama koje se mogu pojaviti u budućnosti.

Mogućnost izvoza i uvoza podataka, osim što omogućava jednostavan unos velikog broja korisnika, spremanje korisnika van sustava ili prijenos, nudi i dobar izvor podataka za analizu različitih karakteristika bivših studenata fakulteta.

Jurica Zrna

8. Literatura

1. Christoph Haas, ISPmail tutorial for debian squeeze, URL: <http://workaround.org/ispmail/squeeze> (19/5/12)
2. The Django Book, URL: <http://www.djangobook.com/en/2.0/> (18/6/12)
3. Redirect Request to SSL, URL: <http://wiki.apache.org/httpd/RedirectSSL> (23/6/12)
4. David Pashley, Becoming a X.509 Certificate Authority, URL: <http://www.davidpashley.com/articles/cert-authority.html> (30/6/12)
5. Django documentation, URL: <https://docs.djangoproject.com/en/1.4/> (13/5/12)
6. Apache, URL: <http://wiki.debian.org/Apache> (3/7/12)
7. Postfix Documentation, URL: <http://www.postfix.org/documentation.html> (18/5/12)
8. regex - What regular expression will match valid international phone numbers? - Stack Overflow, URL: <http://stackoverflow.com/questions/2113908/what-regular-expression-will-match-valid-international-phone-numbers> (22/6/12)
9. John W. Shipman, Python XML processing with lxml, URL: <http://infohost.nmt.edu/tcc/help/pubs/pylxml/web/index.html> (4/9/12)
10. Kevin van Zonneveld, Schedule tasks on Linux using crontab, URL: http://kevin.vanzonneveld.net/techblog/article/schedule_tasks_on_linux_using_crontab/ (1/9/12)