**Razredni diagrami**

**Kontoleri**

Kontroleri služe za dohvat odgovarajućih podataka iz modela te prikazivanje pogleda

povezanih uz danu akciju, u projektu imamo slijedeće kontrolere:

* **Application controller**
  + Glavni kontroler cijele aplikacije.
  + Prikazuje layout u kojem se prikazuju ostali kontroleri.
  + Korisnici i posjetitielji ne vide niti mogu utjecati na proces izvođenja ovog

kontrolera.

* + Sadrži metode koje moraju biti globalno dostupne.
  + Provjerava dali je korisnik prijavljen u sustav.
  + Vraća trenutnog korisnika.
  + Ispisuje HTML tagove za Faviconu, te za ikonu za socijalne mreže.
  + Dohvaća URL gravatar slike trenutnog korisnika.
  + Vraća puni naslov stranice.
* **Users controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled korisnika u sustavu.
  + Omogućuje postojećem korisniku, ako ima privilegije za to, da stvori, izmjeni
  + ili pregleda profil drugog korisnika.
* **Categories controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled kategorija.
  + Kontroler kategorija ima kozmetičku svrhu. Namjena mu je da
  + omogući grupiranje stranica u kategorije.
  + Registriranom korisniku nudi mogućnosti editiranja, brisanja i stavaranja

novih kategorija, te prikaz stranica unutar neke kategorije.

* + Pošto se kategorije obično organiziraju u strukture stabla model

kategorija sadrži polje “ancestry” u koje ancestry gem upisuje organizaciju

stabla.

* **Networks controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu socijalnih mreža.
  + Nema funkcionalnu, već kozmetičku svrhu.
  + Omogućuje dodavanje poveznica na socijalne mreže, ali samo

ovlaštenim korisnicima.

* **Statements controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled prometa.
  + Nudi opciju prikaza dnevnog, mjesečnog i godišnjeg prometa ovlaštenom

korisniku.

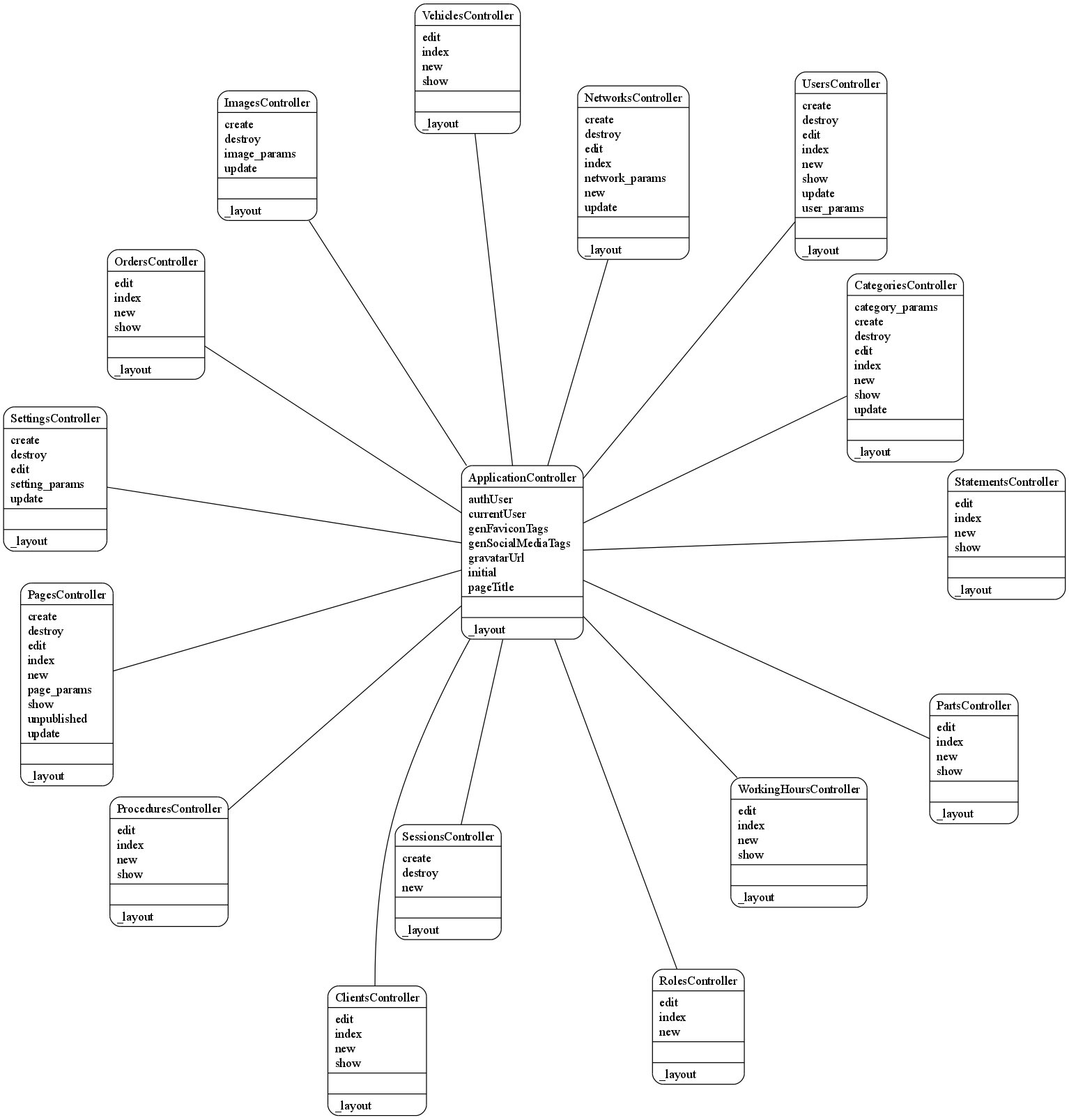
* **Parts controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled dijelova.
* **Working Hours controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu radnog vremena.
* **Roles controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu uloga.
* **Sessions controller**
  + Sadrži metode za stvaranje i brisanje session kolačića u koji je upisan

autentifikacijski ključ.

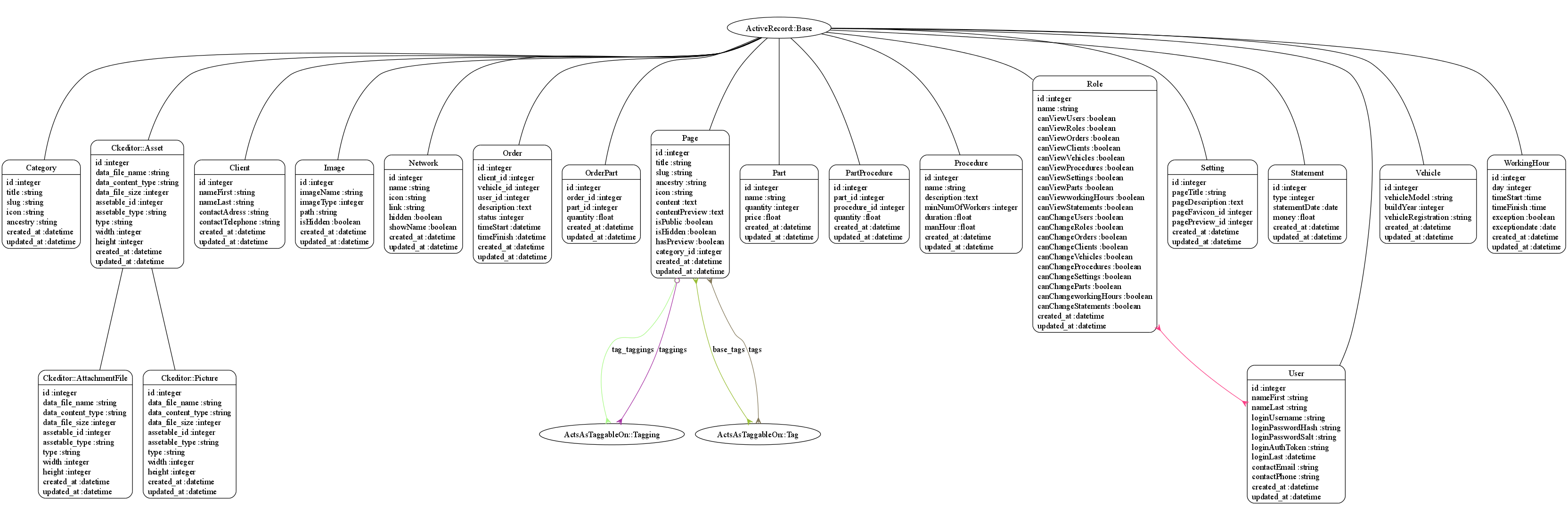
* **Clients controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled klijenata.
* **Procedures controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled postupaka.
* **Pages controller**
* Sadrži metode za stvaranje, brisanje, izmjenu i pregled stranica.
* Nudi opciju pretraživanja stranica po njihovim tagovima.
* Dodatno još nudi opciju prikaza sakrivenih stranica i neobjavljenih

stranica.

* **Settings controller**
  + Sadrži metode za izmjenu postavki.
* **Orders controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled narudžbi.
* **Images controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu slika.
* **Vehicles controller**
  + Sadrži metode za stvaranje, brisanje, izmjenu i pregled vozila.



Slika 1.1



Slika 2.2

**Modeli**

Model je sloj u kojem se nalazi cjelokupna podatkovna logika. S obzirom da je

model podataka detaljno opisan u opisu baze podataka, on ovdje neće biti detaljno

razmatran.

Važno je za napomenuti da u RoR aplikacijama svaki model sadrži  
polje id tipa integer, polje created\_at tipa datetime, te polje updated\_at tipa datetime.

Ako nije suprotno navedeno, pretpostavlja se da svaki model sadrži navedena polja.

* Category

Category je model koji pohranjuje kategorije stranica.  
Sadrži polja title, slug, icon, ancestry.  
Title je tipa string i u njega se pohranjuje naslov kategorije.

Slug je tipa string i u njega se pohranjuje željeni URL stranice.

Icon je isto tipa string i u njega se pohranjuje HTML kod za prikaz ikone

koja će se postaviti prije naslova kategorije, ako je to moguće.

Ancestry je tipa string i služi za organizaciju podataka u stablo, te ga zahtjeva

gem Ancestry

* Ckeditor::Asset

Ckeditor modeli nisu stvoreni ručno već automatski od CKEditor gema

te zbog toga smatram da ih nije potrebno detaljno objašnjavati.  
Koristi se za pokranjivanje priložaka tekstu koji su dodani kroz CKEditor.

Sadrži ime, tip, veličinu (u bajtovima), širinu i visinu (ako je slika) te uz koji unos

je asociran priložak.

Ovo je apstraktni model.  
Za detaljniji uvid treba se obratiti dokumentaciji CKEditor gema.

* Ckeditor::Attachmentfile

Isto kao i Ckeditor::Asset, ali isključivo za datoteke koje nisu slike.

* Ckeditor::Picture

Isto kao i Ckeditor::Asset, ali isključivo za datoteke koje su slike.

* Client

Client je model koji sadrži podatke o klientima s kojima je radionica poslovala.

Sadrži ime i prezime klijenta u poljima nameFirst i nameLast spremljene kao string.

Kontakt adresu i kontakt Telefon se pohranjuje u polja contactAdress i contactTelephone

isto kao string.

* Image

Image je generični model za pohranjivanje slika.

Svrha mu je spremanje slika od korisničkih profila te od djelova sa lagera.

Ima mogučnost pohranjivanja imena slike u polje imageName kao string.  
Također može pohraniti tip slike u polje imageType kao integer.

Path je najbitnije polje u ovom model jer ono pamti gdje je na serveru slika pohranjena.

isHidden je polje koje omogućava sakrivanje slika, ako to korisnik želi.

* Network

Network je model za pohranjivanje socialnih mreža.

Pamti samo najosnovnije informacije poput imena, ikone i poveznice na socialnu mrežu.

Ime se sprema u polje name kao string.

Ikona se sprema u polje icon kao string koji sadrži HTML kod za generiranje ikone.

Poveznica na smau socialnu mrežu se sprema u polje link kao string.

Polje hidden je tipa boolean i omogućuje skrivanje socialne mreže od posjetitelja, u slućaju

da ju ne želimo prikazivati na stranici, ali ju ne želimo niti obrisati.

showName je isto tipa boolean i omogučuje prikaz imena socialne mreže uz njezinu ikonu.

* Order

Model Order u sebi pohranjuje informacije o pojedinačnim radnim nalozima.

client\_id pohranjuje identifikacijski broji (integer) klijenta koji je zatražio zahvat.  
vehicle\_id pohranjuje identifikacijski broj (integer) vezila koje je klijent prijavio za zahvat.

user\_id pohranjuje identifikacijski broj (integer) referenta zaduženog za radni nalog.

Polje description pohranjuje opis željenog zahvata kao formatirani string (niz znakova).

Polje status bilježu u kojem je stanju radni nalog ( otvoren, storniran, … ) kao integer.

timeStart i timeFinish u sebi pohranjuju vrijeme i datum početka i završetka radnog naloga.

* OrderPart

OrderParts je pomočni model koji služi kao posebna vrsta join tabela u to-mani odnosu.  
U RoR aplikacijama je moguće napraviti join table bez modela, ali ako se žele pohranjivati

dodatni podaci u join table onda se za njega mora stvoriti model, ali ne i kontroler.

Pošto moramo pamtiti koliko je dijelova bilo potrošeno u nekom radnom nalogu treba nam

ovaj pomoćni model kako bi smo to mogli pohranjivati u sustav.  
order\_id sadrži identifikacijsti broj radnog naloga, a part\_id identifikacijski broj dijela.  
Polje quantity sadrži kolićinu potrošenog djela. Za tip ovog polja smo se odlučili za float

radi omogućavanja lakšeg proširenja, odnosno ako se korisnik odluči unesti kolićinu tekućine sustav mu to omogućava bez velikih preinaka.

* Page

Page model pohranjuje pojedinačne webstranice.

Naslov stranice pohranjuje u polje title kao string.

URL stranice, u polje slug kao string.

Ikonu, u polje icon kao HTML string.

Sadržaj, u polje content kao HTML formatirani string.

contentPreview je polje za potencijalna proširenja sustava, predviđeno je za dodavanje

kratkog opisa stranice kako se korisniku nebi morala učitavati cijela stranice već samo njezin opis ili primjer.

isPublic je boolean koji odrađuje dali ne registrirani korisnik može pristupiti ovoj stranici.

isHidden je boolean koji odrađuje dali se stranica prikazuje u popisu stranica.  
Važna napomena! Ne registrirani korisnici i dalje mogu pristupati sakrivenim stranicama ako znaju njihov URL.

hasPreview je boolean koji odrađuje dali da se u popisu stranica koristi kratki opis ili puna stranica.

category\_id pohranjuje kategoriju kojoj stranica pripada.

Ancestry je polje koje zahtjeva gem Ancestry i njegova je svrha da organizira stranice u

stabla, međutim postoji mogučnost da će ova funkcionalnost biti izbaćena u novijim verzijama.

* Part

Part je model koji pohranjuje pojedinačne djelove.

Polje name sadrži ime djela kao string.

Quantity sadrži kolićinu djelova istog tipa kao cijeli broj, međutim u novijim verzijama će to biti promjenjeno u tip float kako bi se mogle u sustavu prikazivati tekućine ili slićni materijali.

Polje price sadrži jediničnu cijenu pojedinog dijela.

* PartProcedure

PartProcedure je pomoćni model koji služi kao join table.  
Svrha njegovog postojanja je da pohranjuje kolićinu korištanih dijelova nekog tipa u određenm postupku.  
part\_id i procedure\_id pohranjuju identifikatore djela odnosno postupka, a quantity pohranjuje količinu utrošenog dijela u postupku, kao flaot.

* Procedure

Procedure je model koji pohranjuje postupke.

Ime postupka pohranjuje se u polje name, a opis postupka se pohranjuje u polje description kao string, odnosno HTML string.

minNumOfWorkers je polje koje pohranjuje minimalan broj zaposlenika potrebnih da se postupak obavi. Polje duration pohranjuje trajanje samog postupka u satima.

manHour je polje koje pohranjuje cijenu čuvjek sata za ovaj postupak.

* Role

Role je model koji pohranjuje uloge korisnika u sustavu.  
Jedan je od najbitnijih modela jer određuje koji korisnik ima kakva prava pristupa u programu. Sadrži polje name u koje se pohranjuje ime uloge kao string i dva booleana za svaki kontroler u aplikaciji koji određuju dali korisnik može dodavati, izmjenjivati i brisati u tom kontroleru, odnosno dali smije gledati sadržaj kojim raspolaže taj kontoler.

* Setting

Setting je model koji pohranjuje postavke stranice.

Sadrži ime stranice u polju pageTitle, kao string.  
Opis stranice u polju pageDescription kao formatirani string.

Te identifikatore slika koje će se koristiti za favikonu stranice odnosno za sliku na socialnim mrežama, pohranjene u poljima pageFavicon i pagePreview.

* Statement

Statement je model koji sadrži dnevne mjesečne i godišnje izračune.

U polje type spremamo vrstu izračuna, 0 za dnevni, 1 za tjedni, 2 za mjesečni i 3 za godišnji. To je bitno radi optimizacije izrade izračuna i optimizacije pretraživanja izračuna.

statementDate pohranjuje datum izrade izračuna. Polje money sadrži izračunatu bilansu.

* Vehicle

Model Vehicle sadrži vozila klienata.  
vehicleModel pohranjuje marku automobila kao string.

Godina izgradnje vozila je zapisana u polju buildYear kao integer.

Registracija je zapisana u polju vehicleRegistration kao string.

* WorkingHour

WorkingHours je modle koji sadrži radno vrijeme radionice.  
Model je izrazito bitan za izradu izračuna jer se oni moraju raditi na kraju radnog vremena, koje je određeno ovim modelom.

Polje day sadrži dan u tjednu na koji se radno vrijeme odnosi, 0 za ponedjeljak, 1 za utorak itd. timeStart i timeFinish su polja u koja se pohranjuje vrijeme otvaranja i zatvaranja radionice. Polje exception određuje dali je unos izvanredan te se neće rotirati kao ostala radna vremena u sustavu već će se pojaviti samo na datum iznimke i nikada više ( Ovo je potrebno za praznike i posebne okolnosti ). exceptiondate pohranjuje datum iznimke.

* User

User je definitivno najvažniji model u sustavu. On sadrži sve podatke potrebne za funkcioniranje korisnika u sustavu i omogučuje postojanje korisnika!

Ime i prezime korisnika us pohranjeni u poljima nameFirst i nameLast kao string.  
Korisnićko ime korisnika se pohranjuje u polje loginUsername kao string.  
Korisnikova lozinka se pomoću bycript gema razlaže u njezinu hash i salt komponentu koje se pohranjuje u loginHash i loginSalt poljima ovog modela kao stringovi.  
loginAuthToken je polje koje sadrži nasumično generirani unikatni autentifikacijski token.  
token ima dvije svrhe, za jednoznačnu identifikaciju korisnika iz session kolaćića sa korisnikovog pretraživaća, te za mogućnost potencijalnog proširenja sustava i stvaranje eksternog APIja za komunikaciju sa aplikacijom. Polje loginLast pohranjuje kada se korisnik zadnji puta prijavio u sustav, a polja contactEmail i contactPhone sadrže email adresu i telefonski broji korisnika.