

Todo Appi

Johdanto

Tarkoituksena on tehdä omaan käyttöön/omasta mielenkiinnosta sellainen "motivationalinen" todo appi nettisovelluksena. Se on siis sellainen, jossa asetetaan esimerkiksi kuukausittaisia tai viikottaisia tavoitteita (esim. "Haluan käydä salilla 3x vk" tässä kuussa tai "Tipaton kuu"), jotka ovat sitten jatkuvasti esillä muistuttamassa. Lisäksi on tietysti päivittäisiä todoja tai tekstipohjaisia "Milan eläinlääkäri" muistutuksia.

Käytetään Flask -web frameworkkia eli ohjelmointikeili on Python ja tietokantana SQLite, palvelu hostataan Herokuun. Sovellukseen tulee todennäköisesti kuulumaan myös jonkin verran Javascriptiä.

Yleinen idea

Näkymä on keskeä kahti halkaistu kuva ("vihkomainen") ja vasemmalle jää motivoivat elementit ja ryhmäjutut ja oikealle itse Todo-näkymä. Käyttäjä voi liittyä ryhmiin, joilla on jotain tavoitteita esim. "Venytellään joka päivä" tai jotain ja se näkyy myös tuossa vasemmalla, jos käyttäjällä on useita ryhmiä hän voi navigoida niitä tästä näkymästä valikolla tai scrollaamalla (ei varmaa).

Sivu jaettu keskeltä kahtia

Yläosa vasemmalla:
Kuukausittaiset

Alaosa vasemmalla: Ryhmät
Jos ei ryhmää, tätä ei näy.

Goal	Motivation
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Group	
Ryhmän title ja kuvaus	
Uusin postaus ryhmään	

Päivämäärä
Note
Note
<input type="checkbox"/> Todo
<input type="checkbox"/> Todo
<input type="checkbox"/> Todo

Päivittäinen

Päivämäärä ja navigaatio

Notet tulevat ensin (jos on)

Jos noteilla on aika (isTimed) niin ne näytetään aikajärjestyksessä esim:

12:00 eläinlääkärit

14:00 luento

Näiden alle tulee Todot:

Suunnilleen näin, navigaatioasioita yläpalkissa (?), esim. Uusien ryhmien haku. Muuten esimerkiksi Goal/Motivation klikkaamalla suoraan editointiin, vastaavasti todojen/notejen lisääminen suoraan tästä näkymästä jollakin [+] symbolilla.

Sivusuunnitelmat

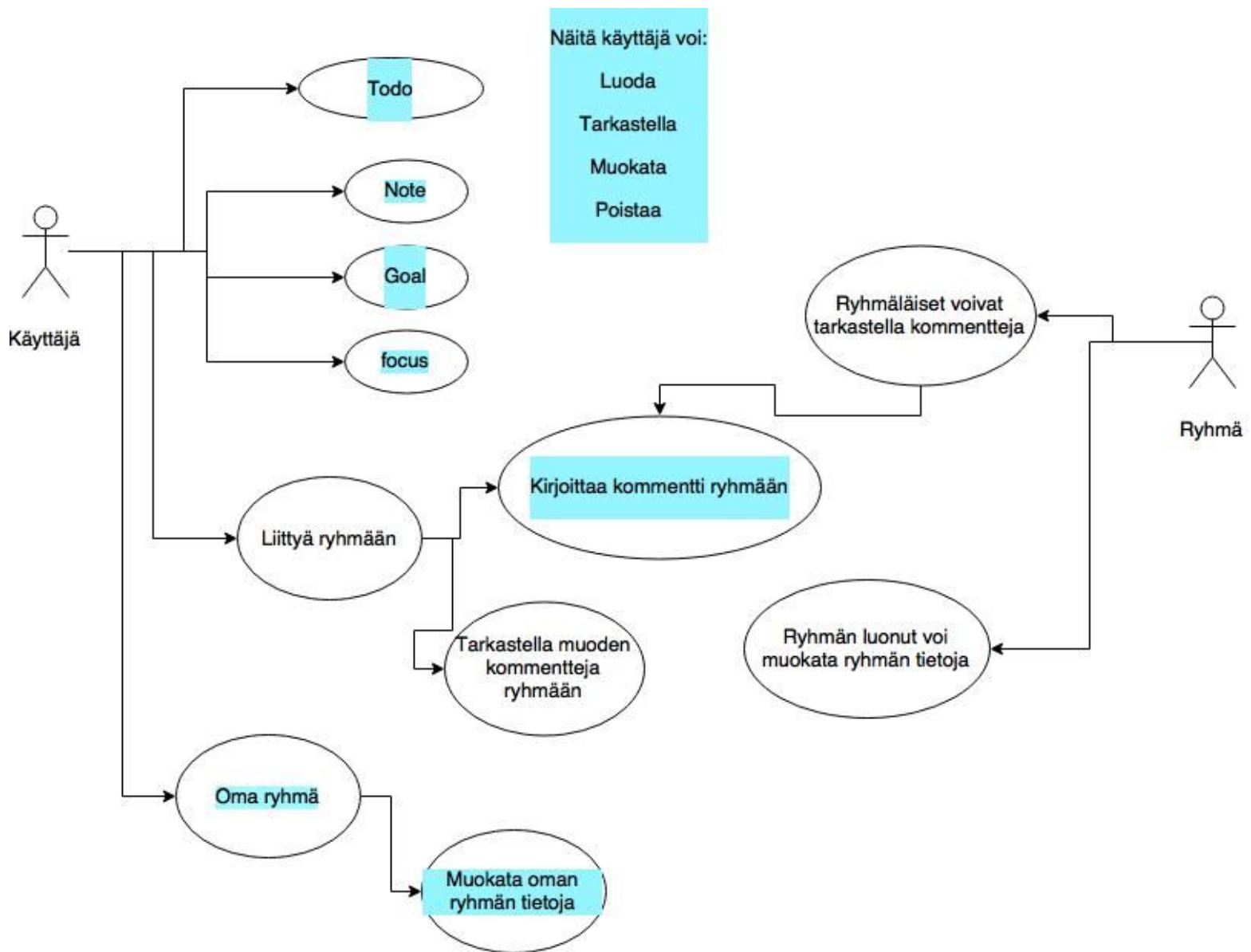
Listaussivu: Kaikki groupit listautuu yhdessä näkymässä jossa näkyy groupin tiedot, kuten nimi ja kuvaus.

Editointisivu: Pieni sivu groupin nimi ja kuvaus.

Lisayssivu: Samanlainen kun editointisivu.

Yleiskuva järjestelmästä

Käyttötapaukset



Käyttäjäryhmät

Käyttäjät ovat kaikki käyttäjiä (User) jotka haluavat tehdä todoja/noteja/ym ja liittyä ryhmiin.

Käyttötapauskuvaukset

Käyttäjä voi lisätä/poistaa/muokata kaikkia omia Todoja/Noteja/Goaleja/Focuksia ja omia kommenttejaan.

Todoja ja Noteja liittyvät johonkin päivään.

Todojen valmistumisesta saa pisteitä.

Focus/Goals saa myös pisteitä.
Noteille voi asettaa ajan.

Ryhmiä voi tarkastella listassa jossa on kaikki ryhmät, joista linkki yksittäisten ryhmien sivuille.

Grouppien hallinnointi:

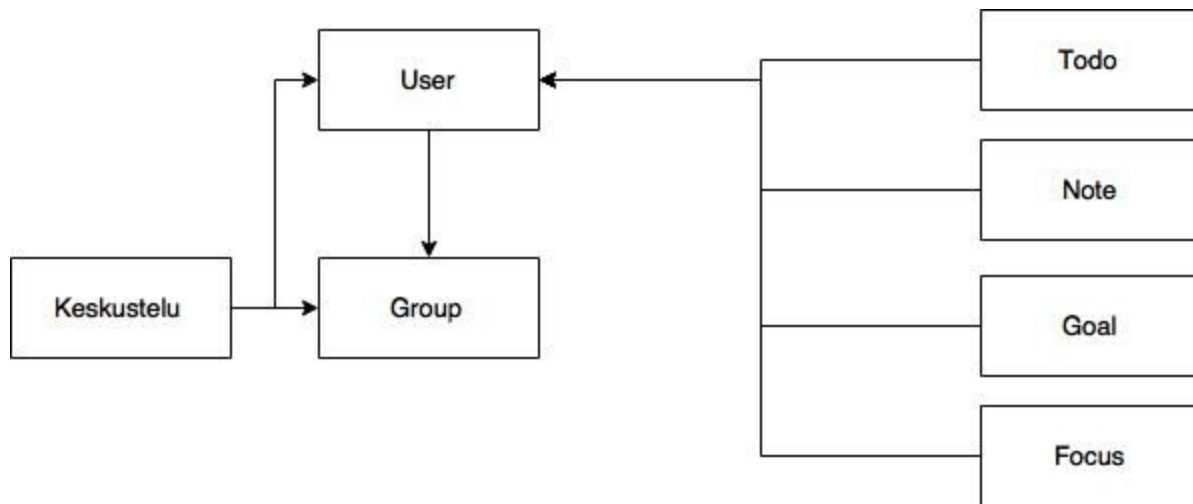
Ryhmän luoneesta henkilöstä tulee admin.

Admin voi asettaa muita adminneiksi tai poistaa ryhmästä.

Ryhmässä kommentointi:

Ryhmään voi kommentoida jos on ryhmän jäsen.

Järjestelmän tietosisältö



Toteutetut:

User

id	Integer, primary key, autoincrement	User ID
email	Text, not null	Käyttäjän Email
name	Text, not null, unique	Käyttäjän nimi
hash/salt	Text, not null	Hashattu salasana
created_at	Datetime	luotu
edited_at	Datetime	Muokattu viimeksi

todos	integer	Kuinka monta todoa tehty
goals	integer	Kuinka paljon goal pisteitä
focus	integer	Kuinka paljon focus pisteitä

Note

id	Integer, primary key, autoincrement	Note ID
user_id	Not null, Foreign key	Käyttäjän ID
text	text	Teksti
isTimed	Bool	Onko ajastettu note
time	Time	Aika
date	Date	Päivämäärä

Todo

id	Integer, primary key, autoincrement	Todo ID
user_id	Not null, Foreign key	Käyttäjän ID
text	text	Teksti
isComplete	Bool	Onko tehty
date	Date	Päivämäärä

Goal

id	Integer, primary key, autoincrement	Goalin ID
user_id	Integer, not null, foreign key	Käyttäjän ID
text	text	Teksti
points	integer	Kerätyt pisteet
isActive	Bool	Onko tällä hetkellä aktiivinen

created_at	datetime	Milloin luotu
edited_at	datetime	Milloin muokattu
end_date	date	Milloin päättyy

Focus

id	Integer, primary key, autoincrement	Focuksen ID
user_id	Integer, not null, foreign key	Käyttäjän ID
text	text	Teksti
points	integer	Kerätyt pisteet
isActive	Bool	Onko tällä hetkellä aktiivinen
created_at	datetime	Milloin luotu
edited_at	datetime	Milloin muokattu
end_date	date	Milloin päättyy

Group

id	Integer, autoinc, primary	Group ID
name	text	Ryhmän nimi
description	text	Ryhmän kuvaus
created_at	datetime	Luonti pvm
edited_at	datetime	Muokkaus pvm
? isDaily	? bool	? ehkä sellainen kenttä joka kuvaa että luodaanko ryhmän kautta päivittäisiä todoja.

User_in_Group

user_id	Integer foreign	Käyttäjän ID
group_id	Integer foreign	Ryhmä ID
isAdmin	bool	Onko käyttäjä tämän ryhmän admin?

GroupDiscussion

id	Integer, primary, autoinc	Keskustelun ID
group_id	Int foreign	Ryhmän ID
user_id	Int foreign	Postanneen käyttäjän ID
title	text	Otsikko
text	text	Teksti
created_at	datetime	luotu
edited_at	datetime	muokattu

Focus_Tag

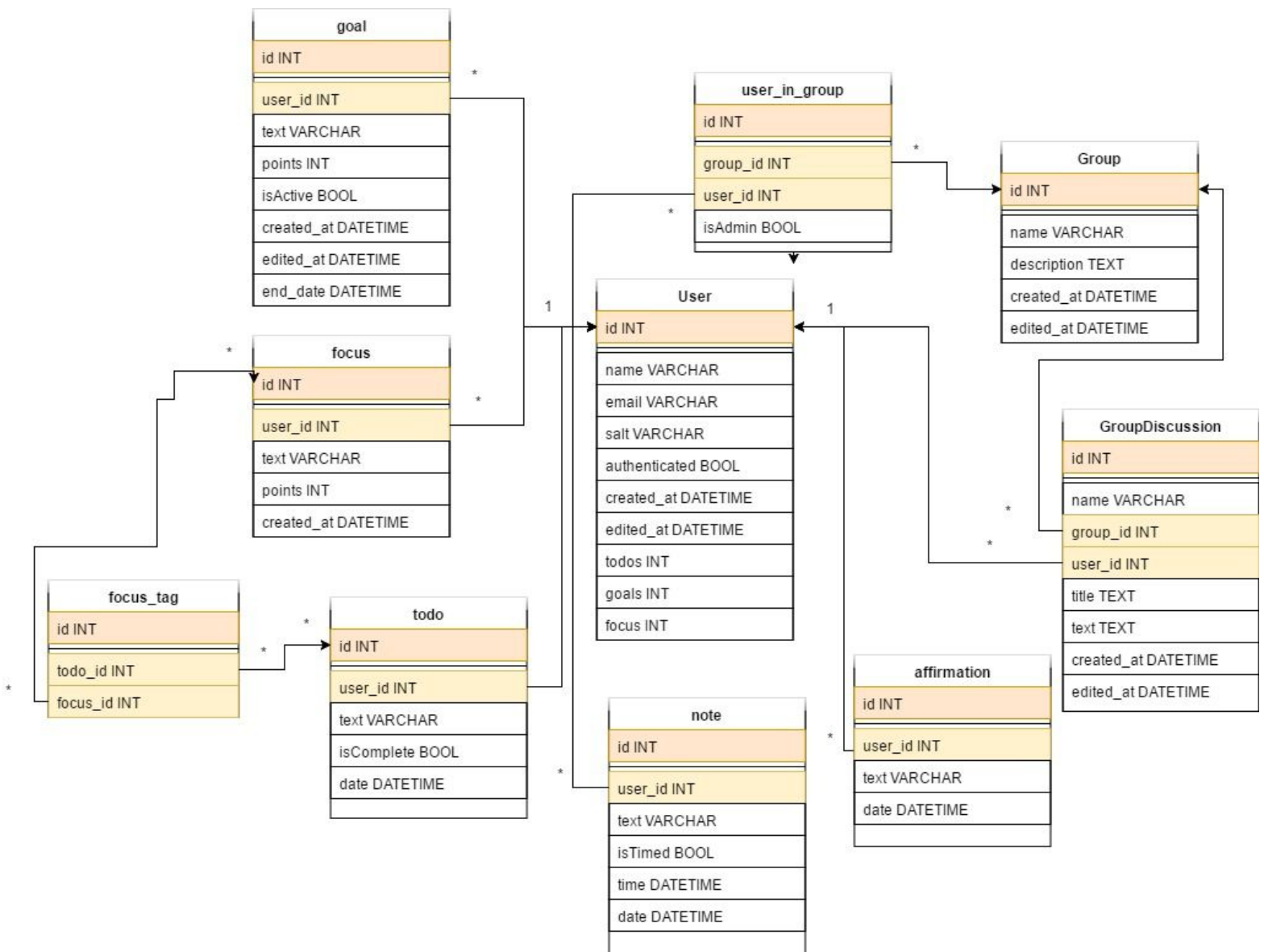
focus_id	Integer foreign	Focuksen ID
todo_id	Integer foreign	Todon ID
id	Integer primary key	Focus Tagin ID

Affirmation

id	Integer, primary key, autoincrement	Todo ID
user_id	Not null, Foreign key	Käyttäjän ID

text	text	Teksti
date	Date	Päivämäärä

Relaatiotietokantakaavio



Järjestelmän yleisrakenne

Tietokantasovellusta tehtäessä on yritetty noudattaa MVC-mallia mahdollisimman hyvin.

Kaikki modelit (tietokantahaut) ovat app-kansion juuressa, nimetty taulun mukaan monikossa. Esim. users.py, groups.py . Lisäksi app-kansion juuressa on forms.py, jossa on määritelty kaikki applikaation formit ja niiden validaatio.

Kaikki paitsi yksi controlleri ovat app/controllers-kansiossa ja nimetty xController, jossa x on tietokannan mukainen taulu, ainoa joka ei ole nimetty näin on views.py jossa on login/create/ym.

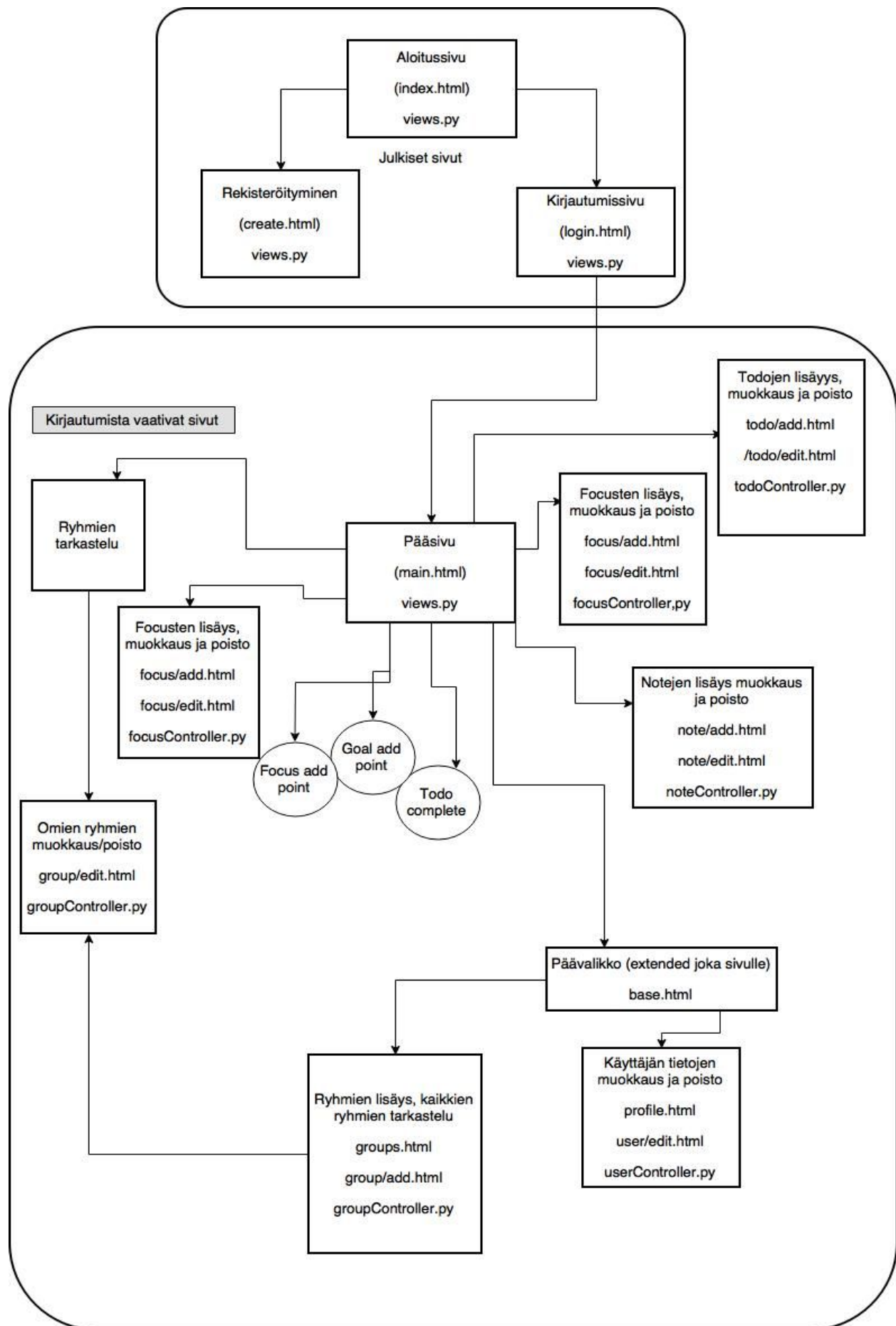
Kaikki viewit ovat templates-kansiossa, joka on vielä alijaoteltu eri taulujen lisäys/editointisivuille, päänäkyä ja login sivut ovat template kansion juuressa.

CSS/Javascript/mahdolliset kuvat ovat static-kansion alla alikansioissa.

Kaikki tiedostonnimet alkavat pienellä kirjaimiella ja jos ovat useasanaaisia muut sanat erotetaan isoilla kirjaimilla esim: noteController.py.

Tietokanta määritellään schema.sql tiedostossa. Sql-kansiossa ne on eritelty niin kun tehtävänannossa pyydettiin.

Käyttöliittymän komponentit



Lähes kaikki toiminta tapahtuu pääsivulla “main”, sieltä ohjataan pääosin vain muokkaus ja lisäyssivuille. Muokkaussivuilta voi poistaa Todon/Focuksen/Goalin yms. Pääsivulta voi myös klikata todot valmiiksi ja lisätä piesteitä goaleihin/focuksiin. Pääsivulla voi liikkua seuraavaan/edelliseen päivään, tässä ohjataan eri kontrollerimetodiin (timetravel), mutta sivu on edelleen main.html ja views.py on edelleen kontrolleri.

Joka sivulle on “extended” base.html jossa on päävalikkopalkki. Päävalikkopalkilla on “Käyttäjän nimi”-valikko johon tulee käyttäjän profiili ja esim. Asetukset. Palkilla on myös “Ryhmät” valikko josta pääsee lisäämään ja tarkastelemaan sekä kaikkia että omia ryhmiä.

Asennustiedot

Requirements.txt sisältää kaikki pakolliset “lisäosat”, joita sovelluksessa on käytetty ja joiden on oltava asennettu. Herokulle asentaessa tämä asentaa kaikki vaadittavat lisäosat. Muuten ne asennetaan pip-python package managerilla:

```
pip install Flask
pip install WTFORMS
...
```

Lokaalisti webserveri käynnistetään ajamalla ./run.py .

Lokaalisti tietokanta luodaan sqlite3 database.db < schema . *

* Viimeiseen palautukseen muutan tietokannan postgressiksi.

Config.py sisältää pakollisia settingejä.

Käynnistys-/ käyttöohje

Sovellus löytyy toimivana osoitteesta:

<https://todoappi.herokuapp.com>

Sisäänkirjautuminen login sivuilta ohjaajalle tunnuksilla:

Username: tsoha
Password: testi1

Tai luomalla uusi tunnus “Create” sivuilta.

Kun käyttäjä kirjautuu sisään, näkyy sivun päänäköymä. Tässä näkymässä voi (+) merkistä aina lisätä kyseisen asian (Note, Todo, Focus, ...), sivu päivittyy. Ylävalikossa on linkki käyttäjän omaan profiiliin ja kaikkiin ryhmätoimintoihin (tosin omat ryhmät näkyvät myös päänäköymässä).

Käyttäjä saa pisteitä kun Todoja tehdään. Ehkä vähän epäselvempi on, että Focus voi olla myös "ryhmä" Todolle, esim jos luodaan uusi focus "Koulu" niin voidaan asettaa Todoja "Koulu" kategoriaan (eli focukseen) ja silloin käyttäjä saa myös focus pisteitä kun tekee todoja. Focuksiin ja grouppeihin saa ihan itsekin lisää pisteitä.

Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Yleisesti ottaen tietty voisi olla hienompi (CSS:ää ja kuvia ja muuta) ja voisi olla "single page application", joka varmaan olisi kuitenkin käyttäjäystävällisyydeltään parempi. Lisäksi ei ole lainkaan sellaisia "konfirmaatiolaatikoita" esim. "Haluatko oikeasti lähteä ryhmästä?", niitä olisi käytettävyyden kannalta hyvä olla.

En ole testannut ohjelmaa muuta kun "käsin".

Parannettavaa:

Lokalisaatio olisi kiva, vaikka kaikki sivujen <title>-tekstit ja valikkotekstit suomeksi/englanniksi.

Osa asioita pitäis varmaan cacheta, jotta ei tarvitse koko aika suorittaa tietokantahakuja ja esim. Päivittää cache vain kun muutoksia tauluun tehdään.

Todo/Note taulut kasvavat tällä tietokantarakenteella lopulta valtaviksi, joten siihen pitäisi olla jokin ratkaisu. Kenties jokin sellainen twittermainen että taulut ovat "päivittäisiä".

Ryhmät

Ryhmissä olisi varmaan hyvä olla salattu/salasanasuojattu mahdollisuus.

Ryhmään pitäisi voida kutsua käyttäjiä.

Keskustelut voisi olla hienommin toteutettuja, esim että liittyisivät toisiinsa eikä vain ryhmään (että voisi kommentoida kommenttia).

Käyttäjille notifikaatio uusista viesteistä ryhmästä.

Ryhmille nimihaku.

Etusivu

Applikaation pääsivun (main) kontrollerissa on aika paljon kaikkea ja siellä on vaikka minkälaisia erillaisia toteutuksia esim. Todojen/Focuksien jne. PUSH-lähetelyihin. Näillä pitäisi olla joku siistimpi ratkaisu tai ainakin yhtenäinen ratkaisu.

Toteutin “kalenterimaisen” ratkaisun niin, että on erikseen “main” kontrolleri tälle päivää ja kun päivä ei ole tämä (datetime.today()) niin käytetään timetravel kontrolleria joka kulkee datetimeä pitkin eteen/taakse. En tiedä onko tämäkään kovin hyvä ratkaisu.

Täällä erikoisesti pitäisi mielummin olla niin, että kaikki todot/jne. Lisätään suoraan tällä sivulla (single page application), eikä mennä eri sivuille. Affirmation toimii esimerkkinä näin.

Käyttäjä

Käyttäjän nimen uniikkiuden validointi tehdään create-kontrollerissa, mutta oikea paikka olisi varmaan forms.py ja WTFFormsin jokin oikea validoija. Tietokannassa tämä kenttä on UNIQUEU

Käytin Werkzeug-nimistä salasanasuojausappia, jossa salasana hashataan ja tallennan hashin tietokantaan. En tiedä onko tämä ihan oikeasti riittävä ratkaisu.

Joku sellainen oikea Admin-käyttäjä joka voisi tarkastella kaikkea olisi varmaan hyvä.

Goals

Tähän ei tullut toteutettua ihan kaikkea mitä tuli mieleen. Oikeastaan tämä nyt suunniteltiin sellaisena epämääräisenä tauluna ilman ihan hirveän selvää suunnitelmaa, mutta siinä oli kuitenkin tarkoitus olla loppumisaika ja sen piti tehdä jotain. Jos nyt pitäisi se tehdä uusiksi niin tekisin sen sellaiseksi, että sen kautta voi asettaa vaikka viikoksi päivittäisen todon tai vaikka viikoittaisen todon tjsp.

Affirmation

Tässä on myös vähän epäselvää mitä haluaisin tällä tehdä, mutta idea nyt oli kuitenkin jonkinlaisten positiivisten ajatusten kirjoittaminen itselle jatkuvasti näkyviksi. Tällaisella positiivisella ajattelulla on ihan motivaation kannalta aito hyöty, mutta toteutus on nyt vähän mitä on ja ei tunnu tarpeelliselta. En tiedä miten tämän saisi sillälailla “esille” ja sellaiseksi, että sitä haluaisi tehdä.

Omat kokemukset

Flask oli oikein kiva ja helppo käyttää. Python nyt muutenkin on todella luonteva kieli, mutta Flaskissa oli todella hyvä dokumentaatio ja varsin “intuitiiviset” komennot. Melkein kaikki Flaskille tehdyt sovellukset jota käytin kuten “Login manager” ja “WTFForms” toimivat suoraan ilman mitään ongelmia tai epäselvyyksiä niiden ihan perus aloitusohjeiden mukaan.

Kirjoitin aluksi tietokantakutsut väärin, Pythonin sen oman string builderin kautta ‘%s’ ym merkeillä. Tämä nyt oli sitten väärin ja vaihdoin ne siihen turvallisempaan ? muotoon. SQL:n

puolelta piti taas pistää mieleen miten ne JOIN-kutsut nyt menikään. Hirveän monimutkaisia kutsuja ei tullut tehtyä. SQL-palautetta käsittelen todennäköisesti vähän huonosti, Python palauttaa listan tupleja ja sen sijaan, että kävisin niitä läpi luettavaan muotoon modelissa ennen palautusta kontrollerille niin käyn ne luettavaan muotoon läpi kontrollerissa eri metodeilla.

Herokun kanssa sai taistella vähäsen, mutta lähinnä omaa hölmöilyä. Mm. väärä versio tarvittavaa lisäosaa requirementsissa joka aiheutti todella outoja virheitä ja kerran vahingossa "siivotessani" poistin oleellisia asioita.

Työn laajuudessa ei varsinaisesti ollut mitään vikaa, tosin olisi ehdottomasti pitänyt tehdä pienempi projekti ja paremmin. Ongelma oli ennemminkin, että en ihan itsekään tiennyt mitä haluan tehdä kaikilla niillä jutuilla mitä lähdin sitten tekemään. Goalit ja Groupit jne. Ovat hyvin kesken ja niissä ei ole kaikkea sitä toiminnallisuutta mitä itse haluaisin. Tarkempi suunnitelma ja idea olisi siis auttanut.