



Santeri Sorjonen, Tatu Pulkkinen, Marko Mandli, Martti Leskinen

# SuperBugTracker -ohjelmisto

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

Toteutusdokumentti

15.10.2021

# Sisällys

1	Johdanto	1
2	Tuotteen vaatimukset	2
2.1	Tarve	2
2.2	Toimintaidea	2
3	Käyttäjäroolit ja käyttötapaukset	4
4	Käsitteet, määritelmät ja ohjelmiston tietomalli	6
4.1	Käsitteet ja määritelmät	6
4.2	Ohjelmiston tietomalli	7
5	Ohjelmiston rakenne	10
5.1	Looginen rakenne	10
5.2	Fyysinen rakenne	11
6	Ohjelmiston toiminta	13
6.1	Käyttäjätilin luominen	16
6.2	Sovellukseen sisäänkirjautuminen	17
6.3	Käyttäjätietojen muokkaus	18
6.4	Ryhmän luominen	19
6.5	Ryhmän hallinta	19
6.6	Projektin luominen	20
6.7	Projektin näkymä ja hallinta	20
6.8	Uuden virheilmoituksen kirjaaminen	20
6.9	Virheilmoituksen hallinta	21
6.10	Kommentin lisääminen	21
6.11	Kommentin muokkaaminen	22
7	Kehitysprosessi ja kehitysvaiheen tekniikat	23
7.1	Kehitysprosessi	23
7.2	Kehitysmenetelmät	23
7.3	Testaus	23
8	Käyttöohje	25

8.1	Sisäänkirjautuminen/Rekisteröinti	25
8.2	Tiimin luonti ja tiimin käyttäjien hallinta	29
8.3	Projektin luonti ja projektien näkymät	32
8.4	Virheilmoituksen luonti ja niihin kommentointi	35
9	Yhteenveto	39

## 1 Johdanto

Tämän dokumentin tarkoituksena on esitellä mahdollisimman laajasti ja kattavasti SuperBugTracker -sovelluksen tämänhetkistä tilannetta ja ominaisuuksia. Myös kehityksessä käytetyt metodit ja ohjelmat käydään yksityiskohtaisesti lävitse. Tavoitteena on, että dokumentin lukemisen jälkeen lukijalla on täsmällinen ja selkeä kuva missä pisteessä projektin kehitys on, ja hän olisi itse kykenevä liittymään projektin kehitykseen vaivatta.

SuperBugTracker on virheilmoitusten seurantaohjelma, jossa painotus on vahvasti ryhmien ja projektien ympärillä. Eli käytännössä käyttäjä voi luoda uuden ryhmän, jonka alle hän voi luoda uusia projekteja, joihin puolestaan voi lisätä uusia virheilmoituksia. Näihin voivat muut käyttäjät, keillä on oikeus nähdä kyseisen projektin virheilmoitukset, käydä kommentoimassa mahdollisia ratkaisuja.

## 2 Tuotteen vaatimukset

Tässä osiossa selostetaan mitä SuperBugTracker tekee ja mitä varten se on kehitetty.

### 2.1 Tarve

Jokaisessa isossa ohjelmointiprojektissa tulee välttämättäkin isoja virheitä, joita yksittäiset kehittäjät eivät löydä, onnistu jäljittämään, taikka osaa ratkaista.

Tästä voi koitua paljon ongelmia projektin etenemiselle, jos virhe on esimerkiksi juuri sellaisessa osassa ohjelmaa, josta monet muut ohjelman osat ovat riippuvaisia. Virheiden ratkaiseminen ja korjaaminen korostuu vielä enemmän, kun on kyseessä suuri organisaatio, jolla on tiukka aikataulu. SuperBugTracker sovelluksen idea syntyi halusta ehkäistä edellä mainittujen ongelmien syntymistä.

### 2.2 Toimintaidea

SuperBugTracker sovelluksen on tarkoitus edistää ja tukea ohjelmistokehitystä. Se on kaikille avoin sekä ilmainen sovellus, joka voidaan julkaisun jälkeen ladata internetistä tietokoneelle. Näin mahdollistetaan, että projektit ja ryhmät, jotka ottavat SuperBugTracker –sovelluksen käyttöön kehityksen tueksi, saavat mahdollisimman suuren käyttajakattavuuden myös virheilmoituksiin.

Sovellus perustuu käyttäjien laatimiin virheilmoituksiin, joita projektien kehittäjät voivat hyödyntää virheiden korjaamisessa sovelluksesta. Joissakin tapauksissa voidaan antaa käyttötukea alkuperäiselle virheilmoituksen kirjoittajalle siitä, miten virheen kanssa tulee edetä. Täten kehittäjät voivat parantaa projektin käyttäjäkokemusta.

Sovelluksessa ei ole vain yksi projekti, jonka alle virheilmoituksia kirjataan, vaan kukin kirjautunut käyttäjä voi luoda ryhmän, jonka alle voidaan luoda niin monta projektia, kuin käyttäjällä on tarve. Ryhmään voidaan myös lisätä muita käyttäjiä, jolloin lisätyt käyttäjät saavat ylläpidollisia oikeuksia ryhmään, sekä sen projekteihin ja niiden virheilmoituksiin. Täten kuka tahansa yksittäinen henkilö, tai

organisaatio voi ladata sovelluksen ja ottaa sen kehityksen rinnalle vaivattomasti.

Sovelluksen käyttöliittymä on suunniteltu mahdollisimman yksinkertaiseksi, jotta käyttäjät, joiden päätavoitteena on lisätä yksi virheilmoitus sovellukseen, saavat tehtyä tämän mahdollisimman helposti. Sillä olennainen osa koko ohjelmiston toimintaa ovat virheilmoitukset, käyttöliittymän suunnittelussa on otettu erityisesti huomioon virheiden luonnin lomakkeen kattavuus, sekä selkeys. Käyttäjälle annetaan selkeät ja kannustavat ohjeet selkeän ilmoituksen luontiin, jotta kehittäjät saavat enemmän informaatiota ilmoituksesta, ja jotta virhe saadaan korjattua tehokkaammin.

### 3 Käyttäjäroolit ja käyttötapaukset

SuperBugTracker –sovelluksessa on neljä käyttäjäroolia. Käyttäjäroolit ovat anonymi käyttäjä, joka ei ole kirjautunut sisään, peruskäyttäjä, joka on kirjautunut sisään, sovelluskehittäjä, eli ryhmän jäsen, sekä ryhmän ylläpitäjä. Käyttäjäroolien käyttötapaukset ovat porrastettuja, joten ylemmän tason käyttäjällä on kaikki käyttötapaukset, jotka häntä alemmalla olevalla käyttäjällä on, sekä myös käyttäjäroolin omat käyttötapaukset.

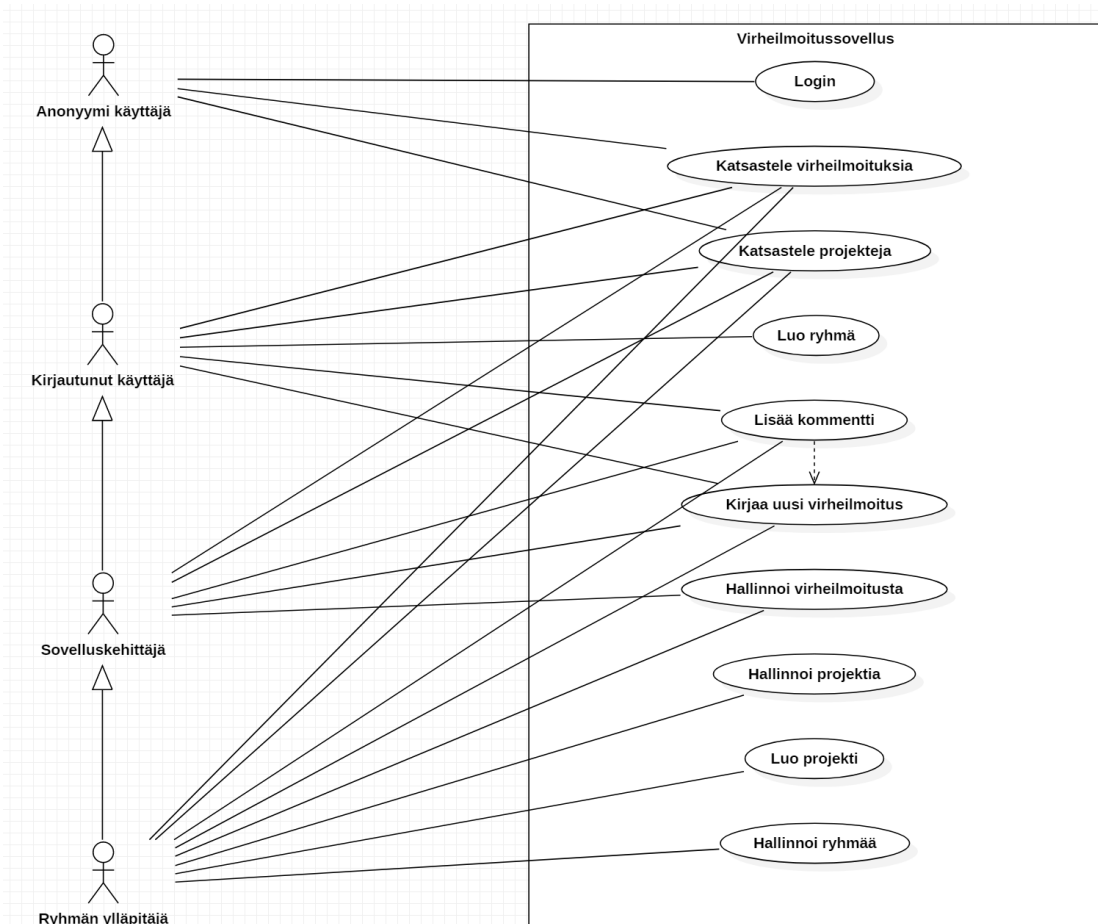
Anonymi käyttäjä voi selata eri ryhmiä ja projekteja, joita sovelluksesta löytyy. Hän voi myös avata virheilmoituksia, ja tarkastella niiden tietoja sekä kommentteja.

Sisäänkirjautunut käyttäjä voi lisätä virheilmoituksia projekteihin, sekä kommentoida virheilmoituksiin. Sisäänkirjautunut käyttäjä voi myös luoda uuden ryhmän, jolloin hänestä tulee kyseisen ryhmän ylläpitäjä.

Sovelluskehittäjä, eli ryhmän jäsen voi hallinnoida virheilmoitusten statusta.

Ryhmän ylläpitäjä on vastuussa ryhmästä, ryhmän jäsenistä, ja sen projekteista. Eli ylläpitäjä voi lisätä uusia projekteja, muokata aikaisempia, sekä halutessaan poistaa projekteja. Ylläpitäjä myös hallinnoi muita ryhmän jäseniä, ja mahdollisesti lisää uusia sovelluskehittäjiä ryhmään, muokkaa heidän rooliaan ryhmässä, sekä tarpeen tullen poistaa ryhmästä jäseniä.

Sovelluksen rakentuessa vahvasti ryhmien ympärille pohjautuvat käyttäjäroolien mahdolliset käyttötapaukset pitkälti siihen, kuuluvatko he ryhmään vai ei. Sovelluksessa on kuitenkin myös mahdollisia käyttötapauksia anonyymeille käyttäjille.



**Kuva 1.** Käyttötapaus kaavio SuperBugTracker –sovelluksesta. Kuva esittää mitä kukin käyttäjärooli voi sovelluksessa tehdä.



## 4 Käsitteet, määritelmät ja ohjelmiston tietomalli

Tämä osio kertoo sovelluksessa käytettävät käsitteet, määritelmät, ohjelmiston tietomallin sekä näiden selitteet. Ohjelmisto sisältää hyvin yksinkertaista sanastoa, jota näiden määritelmien jälkeen on helppo ymmärtää.

### 4.1 Käsitteet ja määritelmät

User – Sovelluksen käyttäjä.

Admin – Tiimin ylläpitäjä ja hallitsija.

Bug/Bugi/Virheilmoitus – Virheilmoitus, joka on sovelluksessa pääosassa. Kuvastaa jotakin ongelmaa ohjelmointiin liittyen enimmäkseen, joka halutaan jakaa sovelluksessa.

Project/Projekti – Sovelluksessa käytettävä sana, joka kuvastaa sovellusta, jota ohjelmoija rakentaa, ja jonka virheitä halutaan selvittää.

Team/Tiimi/Ryhmä – Käyttäjistä koostuva ryhmä.

OPEN – Kuvastaa virheilmoituksen tilaa, johon ei vielä ole saatu toimivaa ratkaisua.

CLOSED - Kuvastaa virheilmoituksen tilaa, jossa ratkaisu on löydetty.

BugState – Ohjelmistossa enum –luokka, joka sisältää edellä mainitut “OPEN” ja “CLOSED” arvot.

Comment – Yksinkertaisesti kommentti, joita voi lisäillä sovelluksessa virheilmoituksiin.

Role - Käyttäjän rooli tiimissä. Sovelluksessa käytetty “Admin” ja “User” rooleja.

Status – Enum-luokka, jota käytetään useassa luokassa attribuuttina ja se kuvastaa tietokantaan lisättävän olion olemassaolon tilan. Kun jokin tieto “poistetaan” tietokannasta sen “Status” muutetaan arvosta “DEFAULT” arvoon “REMOVED”.

UserTeamRoles – Luokka, joka sisältää käyttäjä-, tiimi- ja rooli-olion. Luokan olemassaolon syy on tietokantaan tallennuksessa. UserTeamRoles -tauluun tallennetaan tämän luokan tiedot, joka sisältää tietyn käyttäjän roolin tietyssä tiimissä.

## 4.2 Ohjelmiston tietomalli

Sovellukseen on suunniteltu käytön kannalta oleelliset yksilötyypit, ja suunnittelussa on otettu huomioon, että eri ominaisuuksilla sekä yksilötyypeillä on ‘omistajuus’, eli vain tietyillä käyttäjillä on oikeus niiden kaikkiin käyttötapauksiin.

Käyttäjän yksilötyyppi vastaa käyttäjätiliä, joka on rekisteröity sovellukseen. Käyttäjällä on käyttäjänimi ja sähköposti, salasana ja suola sisäänkirjautumista varten sekä ylläpitäjä -status, joka annetaan sovelluksen ylläpitoa varten.

Tiimi, eli ryhmä koostuu käyttäjistä. Tiimillä on oma tunniste ja nimi. Tiimillä voi olla monia projekteja ja monia käyttäjiä.

Tiiminjohtajat voi luoda tiimille uuden projektin. Projektilla on nimi, jolla sen voi helposti löytää, tunniste, sekä status poistamista varten. Projektilla voi olla allaan monta virheilmoitusta.

Virheilmoitus -tietueet ovat sovelluksen tärkein osa. Sovelluksessa käyttäjä tekee virheilmoituksia, joihin he tarvitsevat tukea muilta käyttäjiltä. Virheilmoitukset voivat sisältää liitteitä, tageja sekä kommentteja. Käyttäjä, joka tekee virheilmoituksen merkitään sen luojaksi. Muita ominaisuuksia virheilmoituksella on mm. päivämäärä, jolloin ilmoitus on luotu, otsikko, jota hyödynnetään, kun sovelluksessa halutaan hakea muiden tekemiä virheilmoituksia, kuvaus, joka

sisältää virheen kuvauksen tekstinä sekä virhetila, joka kertoo, onko virhe ratkaistu.

Rekisteröitynyt käyttäjä voi lisätä kommentteja virheilmoituksiin. Ohjelma lisää kommentin yhteydessä päivämäärän.

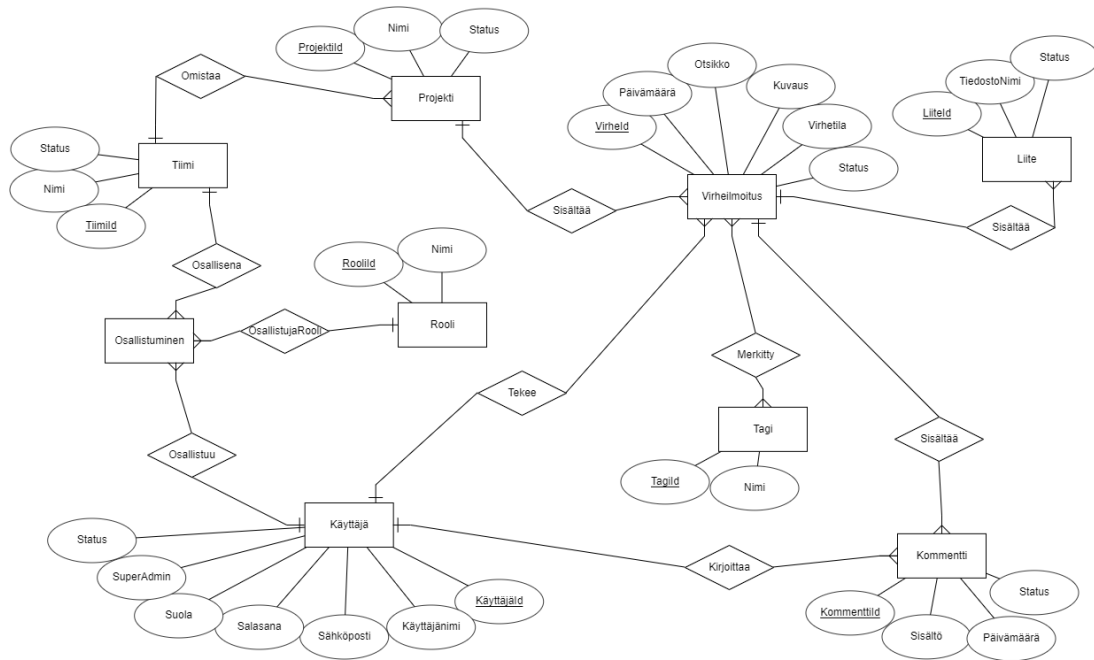
Virheilmoitus voi sisältää liitteitä. Liitteiden tarkoitus on auttaa ymmärtämään virhettä paremmin. Liitteinä voi olla kuvia tai muita tiedostoja. Liitteellä on ominaisuuksina tiedoston nimi, liitteen sisältö binääridatana, status, joka kertoo, onko liite poistettu vai näkyvillä sekä tieto, mihin virheilmoitukseen liite kuuluu.

Virheilmoitukselle voi lisätä kehittäjien luomia tageja kuvaamaan bugin aiheetta. Jokaisella tagilla on tunniste ja nimi, joka sisältää tagin arvon.

Osallistuminen sisältää tiedot käyttäjistä, niiden rooleista ja tiimistä. Yksinkertaistettuna yksilötyypistä nähdään mikä rooli kullakin käyttäjällä on tietyssä tiimissä.

Roolin tarkoitus on kertoa tiimeissä olevien henkilöiden käyttöoikeudet. Esimerkiksi ylläpitäjä -roolilla voidaan hallita tiimin projekteja. Roolin nimi ominaisuus kuvaa mikä käyttöoikeus on kyseessä.

Useissa yksilötyypeissä on status –ominaisuus, jolla korvataan tietueiden poistaminen tietokannasta. Status voi olla joko aktiivinen, tai epäaktiivinen. Jos se on epäaktiivinen, tietue jätetään pois kyselyiden tuloksista.



**Kuva 2.** ER-kaavio SuperBugTracker –sovelluksesta. Kuva esittää yksilötyypit, niiden relaatiot sekä ominaisuudet.

## 5 Ohjelmiston rakenne

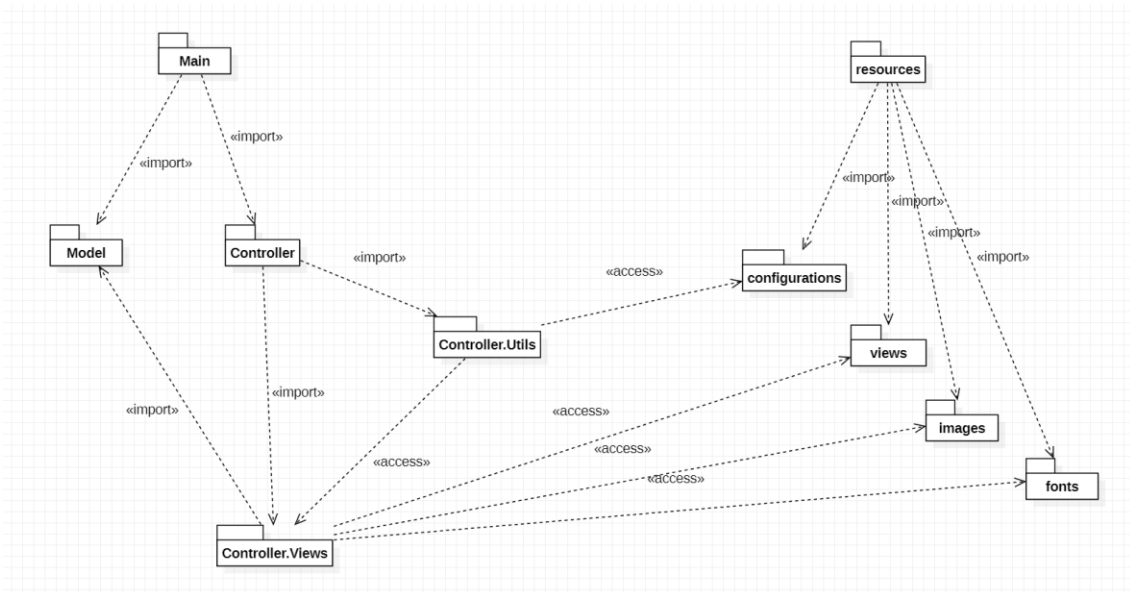
### 5.1 Looginen rakenne

Sovellus on toteutettu käyttäen MVC –mallia (model, view, controller), joten nämä eri luokkatyypit on sijoitettu eri paketteihin. Sovelluksessa on käytetty JavaFX –kirjastoa sekä SceneBuilder –ohjelmistoa näkymien luontiin, joten näkymätiedostot on asetettu “resources” -paketin alle, josta controller –luokat hakevat ne. Ylimmällä tasolla on yksi luokka, MainApplication, jossa on sovelluksen pääohjelma.

Sovelluksessa on useita eri controller -tyyppejä, joista osa on ylemmän tason staattisia controller –luokkia, kuten tietokannan controller, sekä näkymän controller. Kaikilla näkymillä on oma controller –luokka, jossa on tämän näkymän kannalta oleelliset metodit, sekä tiedot. Controller -luokat säilyttävät datan malliluokissa.

Mallipaketissa on kaikki sovelluksen eri tietotyypit omissa luokissaan, joiden tietoja esitellään käyttöliittymässä ja prosessoidaan controller –luokissa. Näitä luokkia myös käytetään tietokantamalleina, eli näiden luokkien tietoja voidaan suoraan lisätä, tallentaa, päivittää tai poistaa tietokantaan.

Resources –paketissa on kaikki sovelluksen käyttämät tiedostot, jotka eivät ole .java -tiedostoformaattissa, eli pääasiassa konfigurointitiedostot, sekä näkymien tiedostot. Myös kuvatiedostot haetaan tästä paketista.



**Kuva 3.** Pakkauskavio sovelluksen tämänhetkisestä tilasta

## 5.2 Fyysinen rakenne

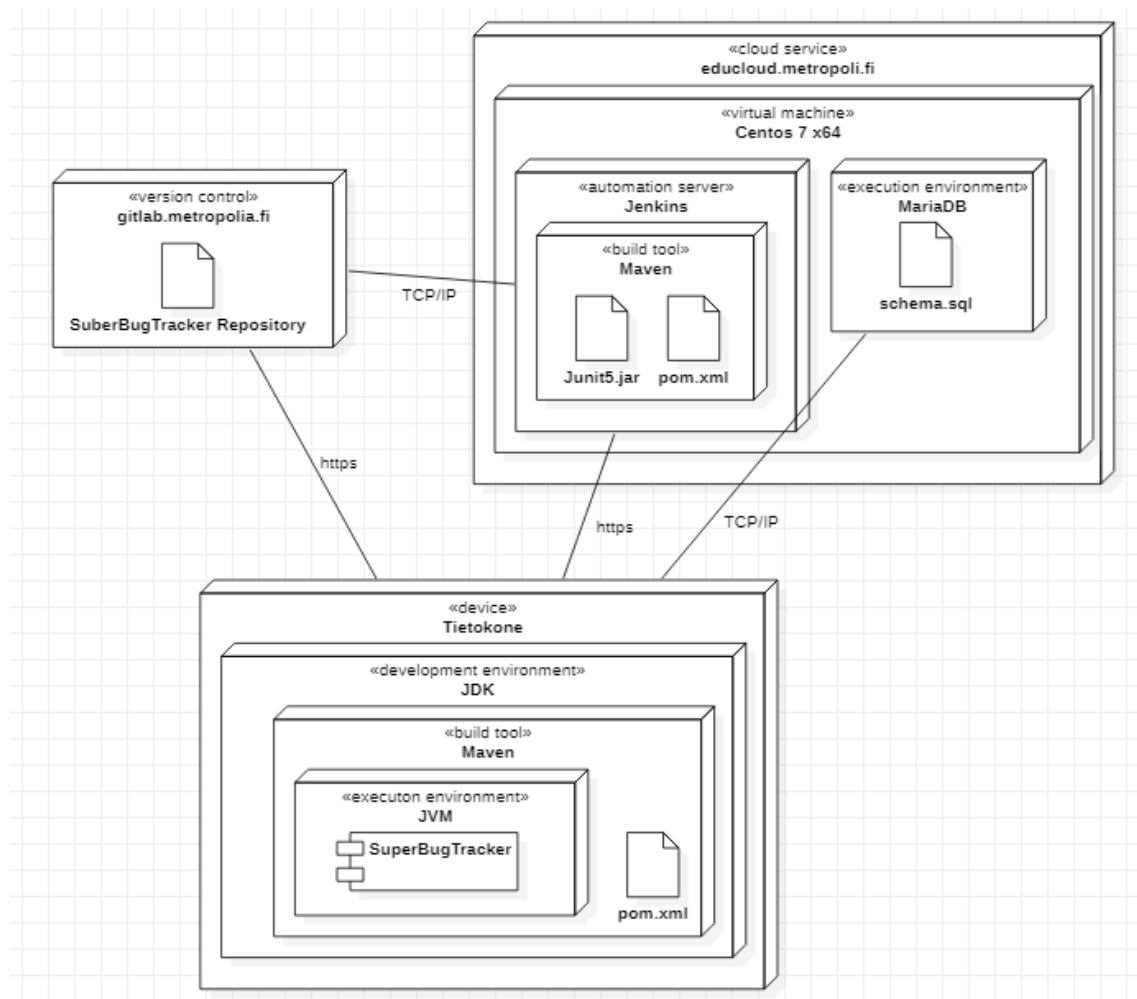
Ohjelmiston rakenne kehitysvaiheessa pohjautuu kolmeen ympäristöön, joista kaksi on pilvipalveluita, ja yksi fyysinen tietokone, joka on projektinjäsenenellä hallussa. Fyysisiä koneita voi olla useita, sillä jokaisella kehittäjällä on oma laite, jota he käyttävät kehityksessä.

Metropolian Educloud –palvelimella on vuokrattu Linux –pohjainen virtuaalitietokone, johon on asennettu Centos 7 x64 –jakelupaketti. Virtuaalikoneen alle on asennettu kehityksen kannalta olennainen automaatiotyökalu Jenkins, jolla varmistetaan koodipohjan testaus, sekä sovelluksen käytön kannalta olennainen tietokantaohjelmisto MariaDB, johon tallennetaan sovelluksen kaikki käyttäjätiedot.

Metropolian Gitlab –palvelimelle on luotu repositorio sovelluksen lähdekoodille, jota sovelluksen kehittäjät käyttävät lähdekoodin versionhallintaan. Myös Jenkins –asennus toisella palvelimella hakee tästä repositorioista lähdekoodin, jonka perusteella ajetaan automaatiotestit.

Geneerisellä tietokoneella sovelluksen kehittäjät kirjoittavat lähdekoodia, sekä ajaa ja testaa ohjelmistoa. Tietokoneelle on asennettu sovelluksen Java

Development Kit -versio, sekä Maven, joilla lähdekoodi voidaan kääntää ohjelmaksi. Sovellus ottaa myös yhteyden pilvessä olevaan tietokantaan, josta sovellus hakee käyttäjädataa. Tietokoneella otetaan myös yhteys Jenkins -käyttöliittymään, jota käytetään verkkoselaimella.



**Kuva 4.** Sijoittelukaavio SuperBugTracker –sovelluksesta.

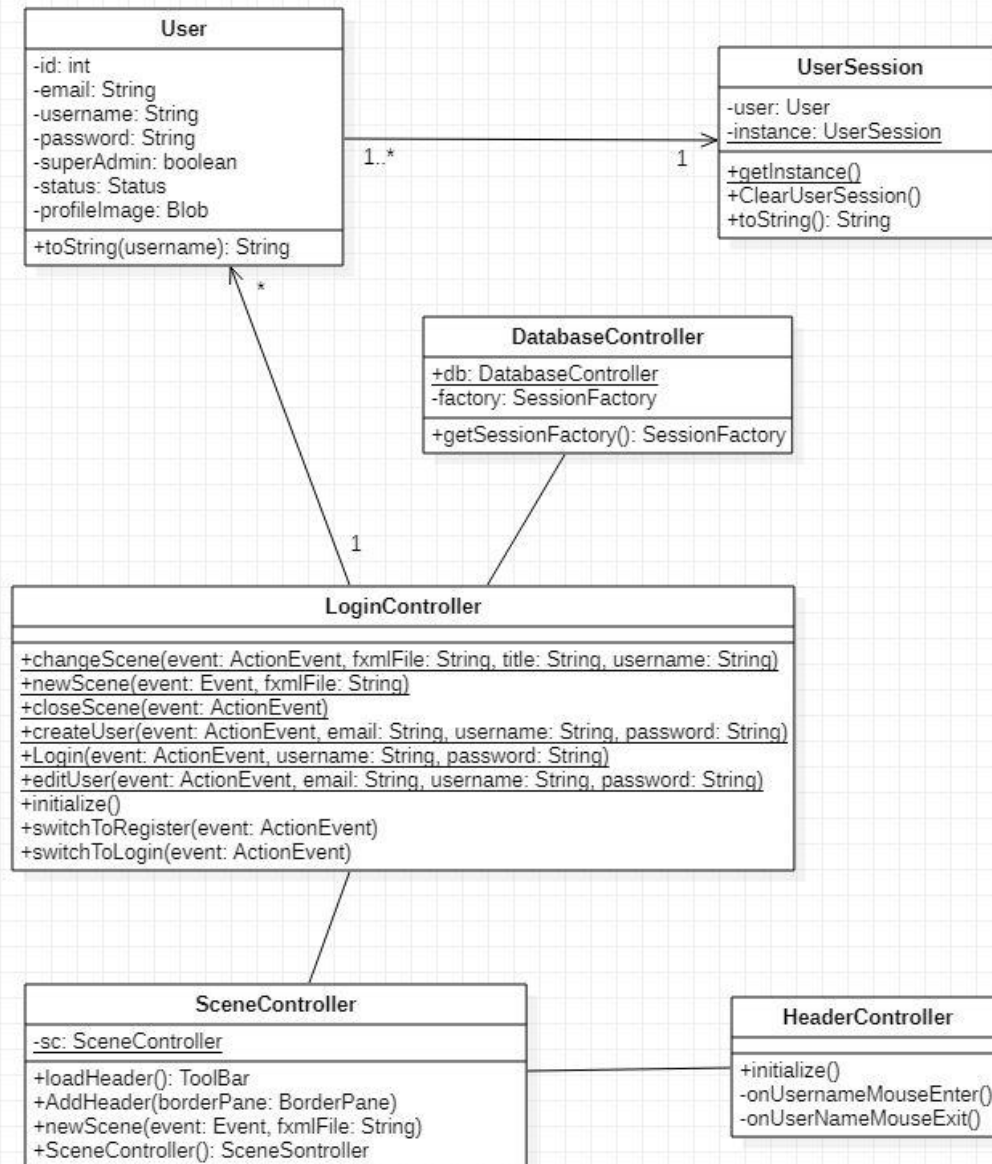
## 6 Ohjelmiston toiminta

Ohjelma saa ulkoiset kirjastot eri palveluista käyttäen Apachen Maven työkalua. Sovellus on kehitetty Oraclen Java Development Kit versiolla 16. Sovelluksen käynnistysluokka on MainApplication, missä luodaan käyttöliittymän pääikkuna, ja luodaan tietokantayhteys.

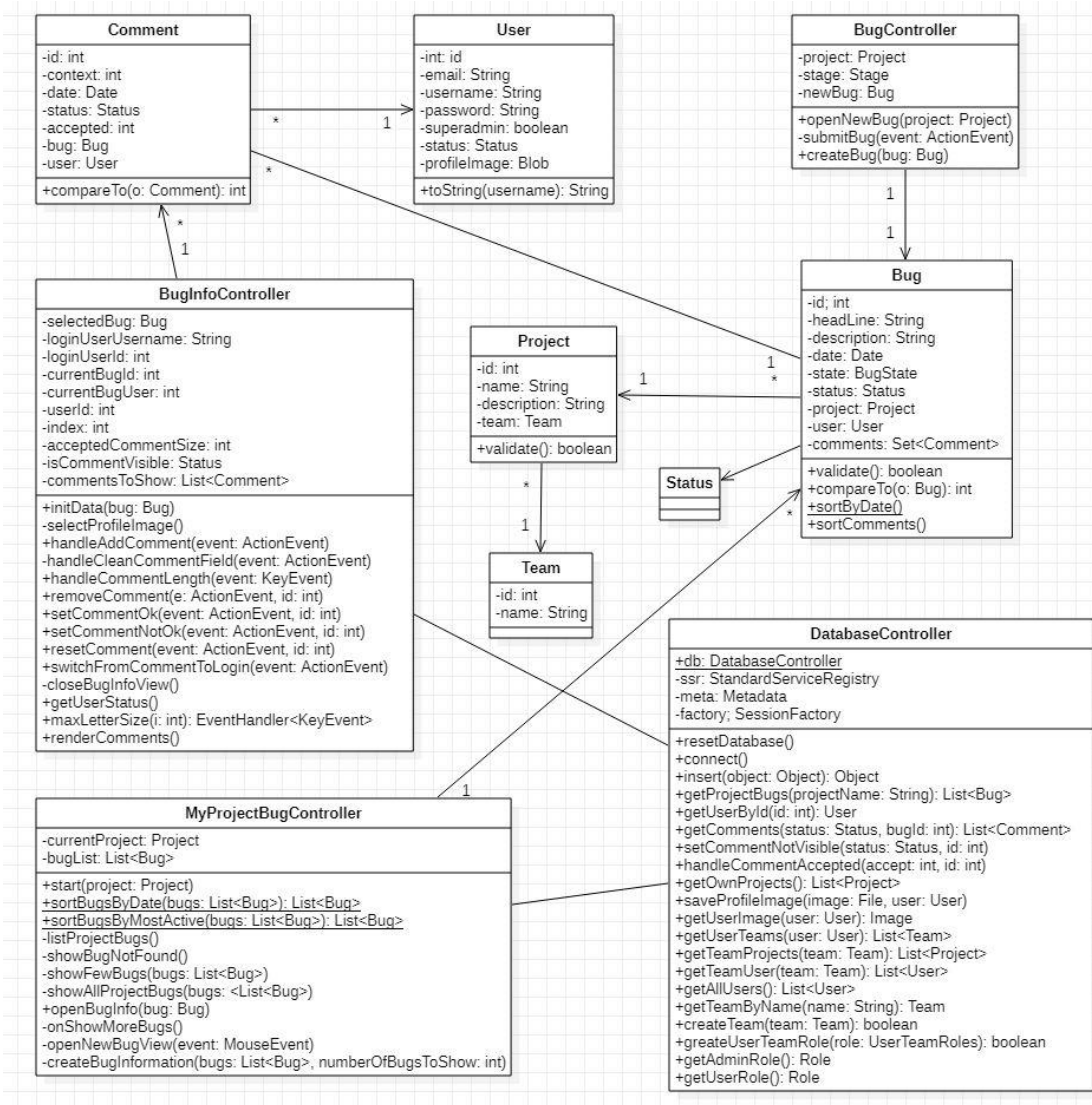
Sovellus käyttää JDBC –kirjastoa tietokantayhteyden luomiseen sekä kyselyiden toteuttamiseen, ja Hibernate –kirjastoa kyselyiden luomiseen sekä yksilötyyppien kartoittamiseen objekteihin. Yhteyden luomisen sekä tietokantakyselyiden metodit ovat luokassa DatabaseController.

Alla ovat luokkakaaviot, jotka esittelevät ohjelman toimintaa.

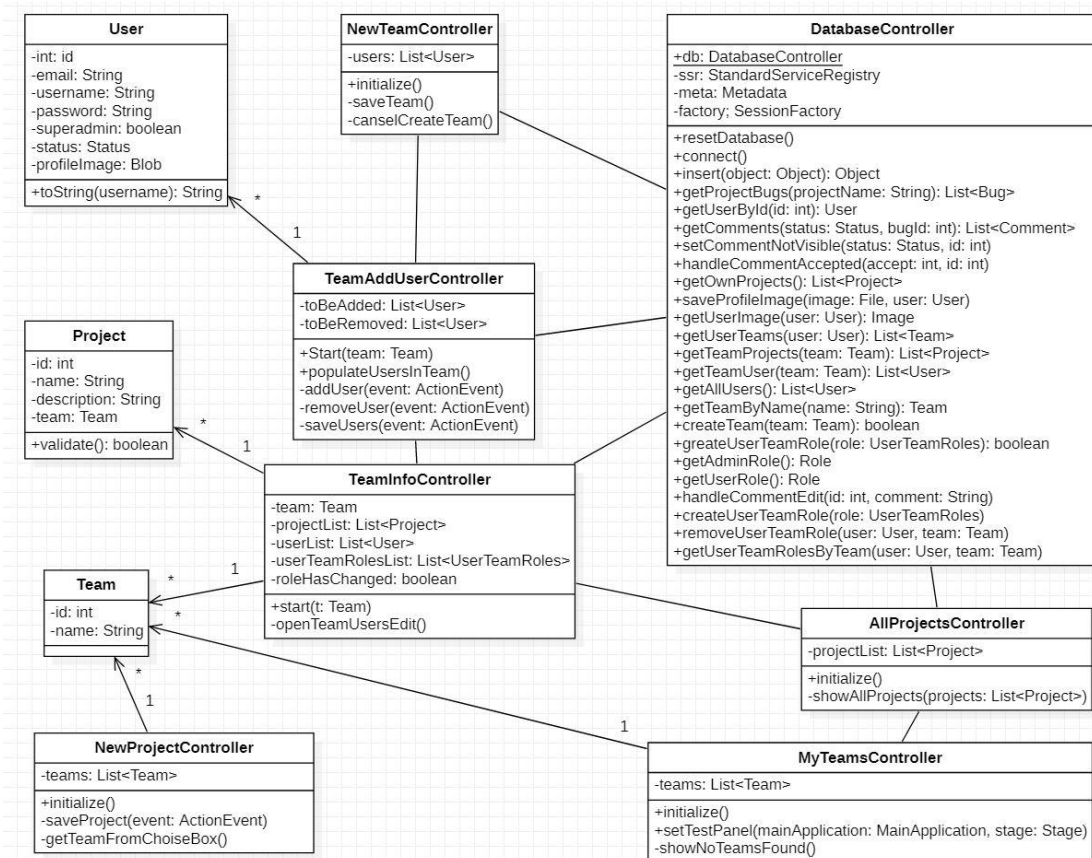




**Kuva 5.** Rekisteröinti, sisäänkirjautuminen, käyttäjätietojen muokkaus



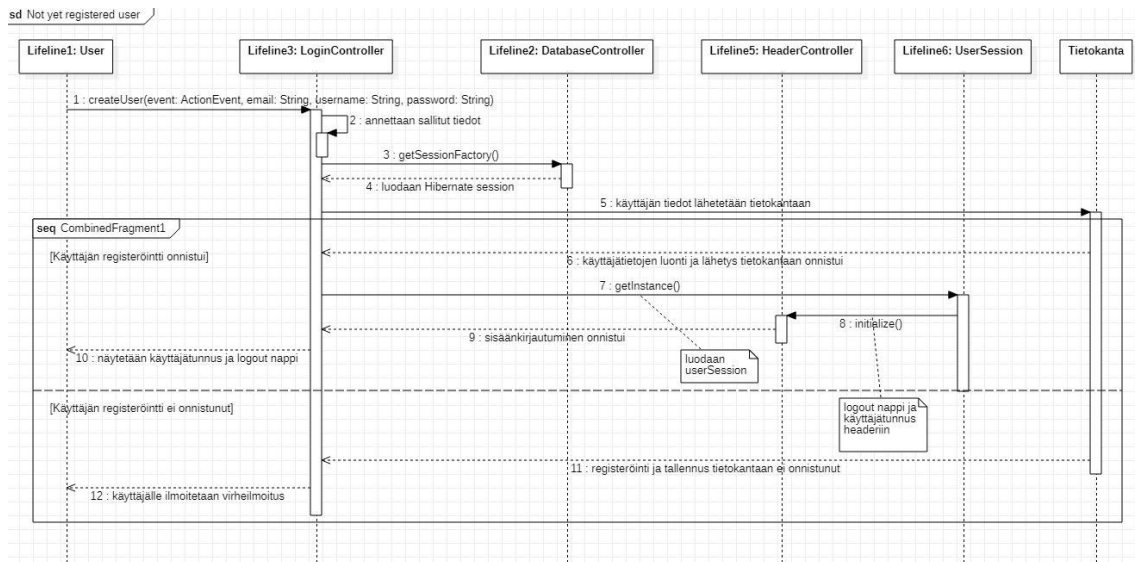
Kuva 6. Virheilmoituksen, kommentin luonti sekä hallinta



**Kuva 7.** Projektin ja ryhmän luonti sekä hallinta

## 6.1 Käyttöjätilin luominen

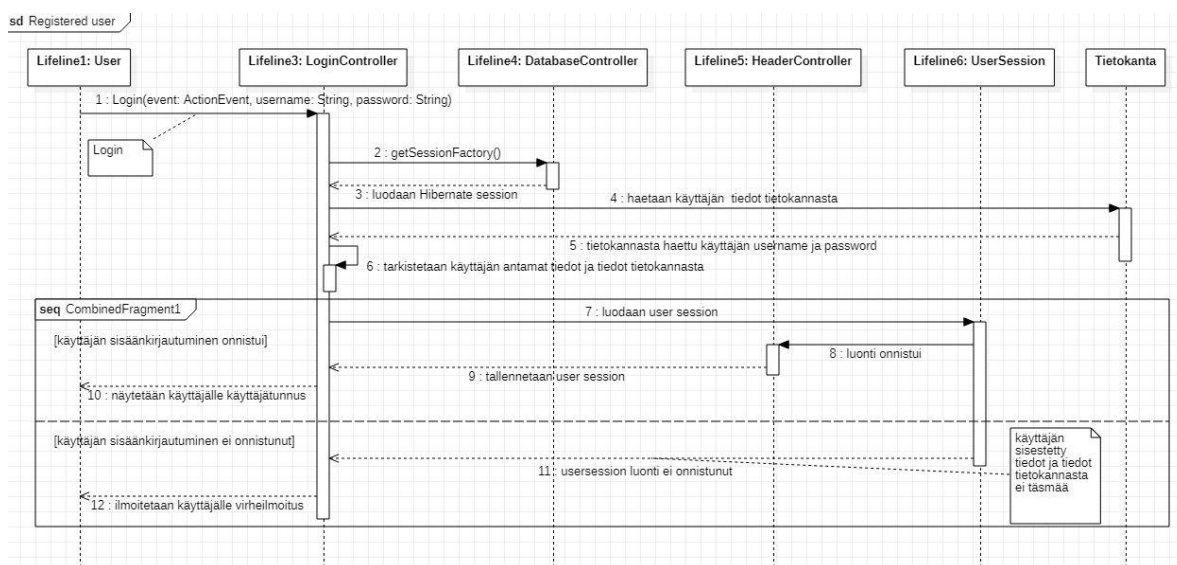
Jos käyttäjä haluaa lisätä virheilmoituksen sovellukseen, täytyy hänen ensin luoda käyttäjätili. Sovelluksen käynnistyessä näytetään pääikkuna, jonka yläpal-kista löytyy sisäänkirjautumis- sekä rekisteröintipainike nimellä "Login". Rekiste-röintilomake on yksinkertainen, ja siinä on vain muutama kenttä, mutta siinä voi silti ilmentyä virhetilanteita, esimerkiksi jos käyttäjänimi on jo tietokannassa, jol-loin ohjelman pitää palauttaa käyttäjälle virheilmoitus. Rekisteröinnin yhtey-dessä käyttäjä myös kirjataan saman tien sisään.



**Kuva 8.** Käyttäjätilin luominen (perustettu luokkakaaviosta: kuva 4)

## 6.2 Sovellukseen sisäänkirjautuminen

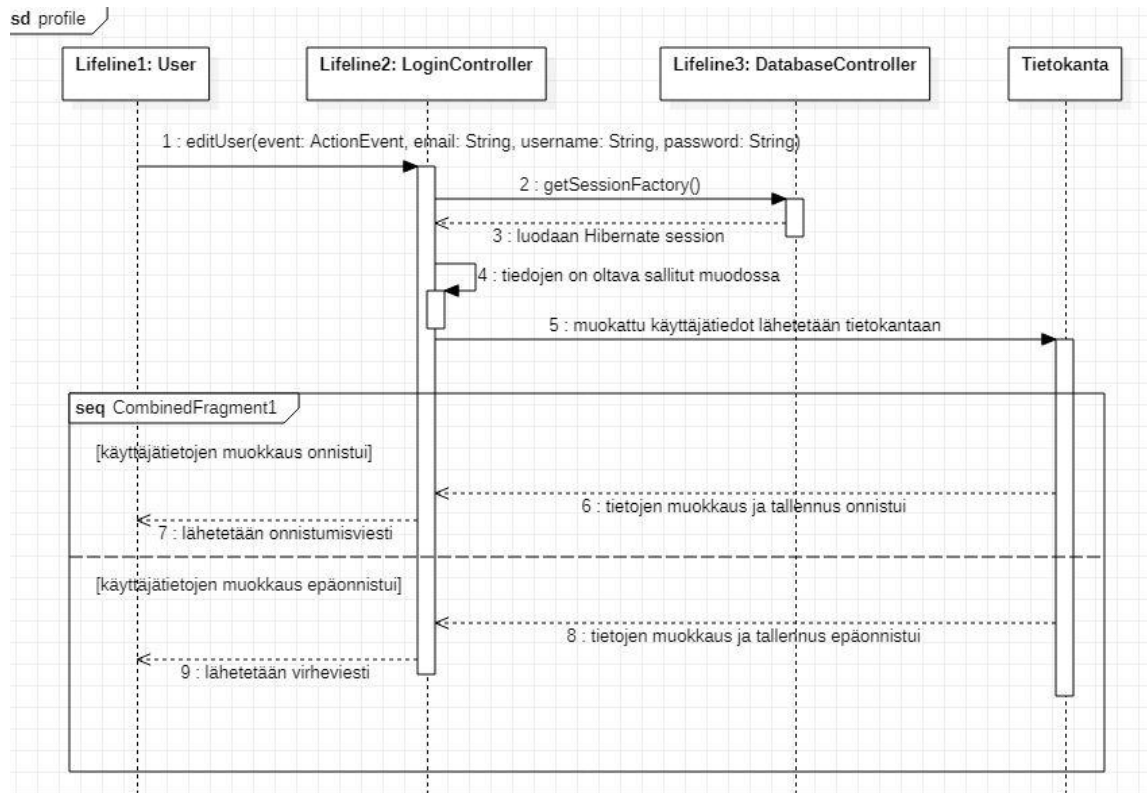
Sisäänkirjautuminen löytyy myös sovelluksen yläpalkista “Login” -napista, johon käyttäjän tulee syöttää käyttäjätunnus sekä salasana. Kirjautumistilanteessa yleisimmät virheet ovat joko käyttäjätunnus, jota ei löydy tietokannasta, tai väärä salasana. Näissä tapauksissa käyttäjälle pitää näyttää virheilmoitus. Kun käyttäjä on kirjautunut sisään, kirjautumisikkuna sulkeutuu automaattisesti ja “Login” -napin tilalle vaihtuu ja “Logout” -nappi.



**Kuva 9.** Sisäänkirjautuminen (perustettu luokkakaaviosta: kuva 4)

### 6.3 Käyttäjätietojen muokkaus

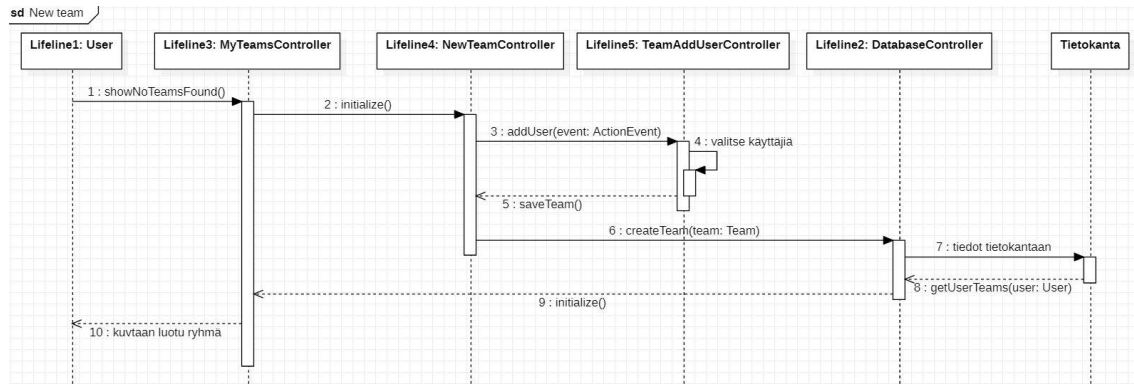
Klikkaamalla käyttäjätunnusta avautuu ikkuna, missä voidaan muokata sähköpostiosoitetta, käyttäjänimeä ja salasanaa. Tiedot validoidaan, ennen kuin ne tallennetaan tietokantaan. Jos tallennuksessa tapahtuu jokin virhe, sovellus näyttää virheviestin käyttäjälle.



**Kuva 10.** Sekvenssikaavio kuvaa käyttäjätietojen muokkaamista ja on perustettu luokkakaaviosta (kuva 4.)

## 6.4 Ryhmän luominen

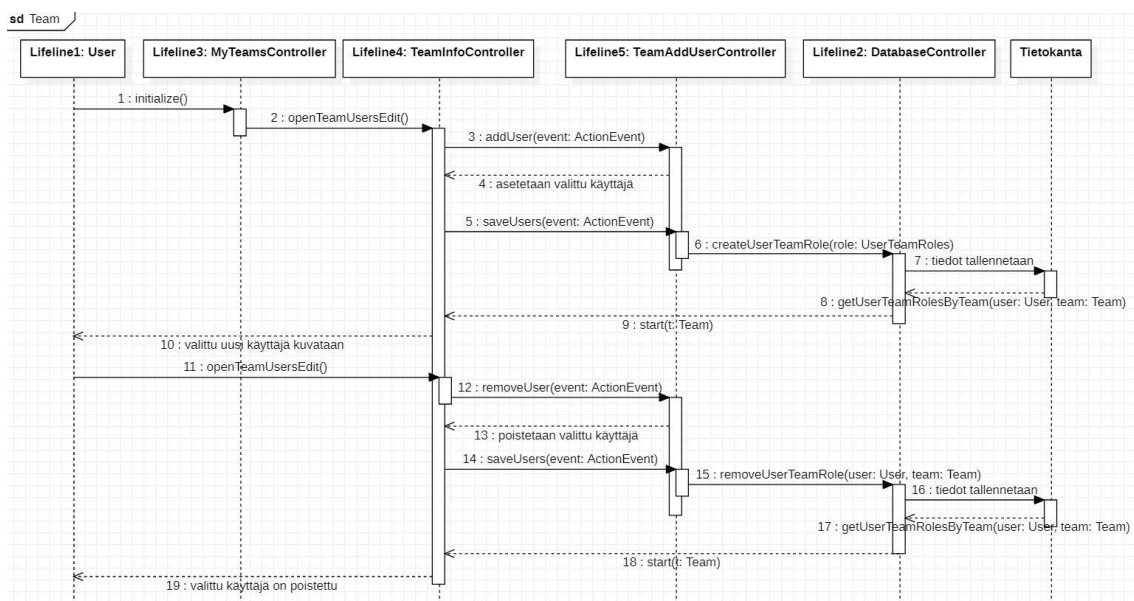
Uuden ryhmän luominen alkaa pääsivulta ja vaatii sisäänkirjautumista. Ryhmän tallennettavat tiedot ovat sen nimi, sekä käyttäjät, jotka kuuluvat ryhmään. Kun ryhmä on tallennettu, käyttäjälle näytetään ryhmän tiedot.



**Kuva 11.** Ryhmän luominen (perustettu luokkakaaviosta: kuva 6)

## 6.5 Ryhmän hallinta

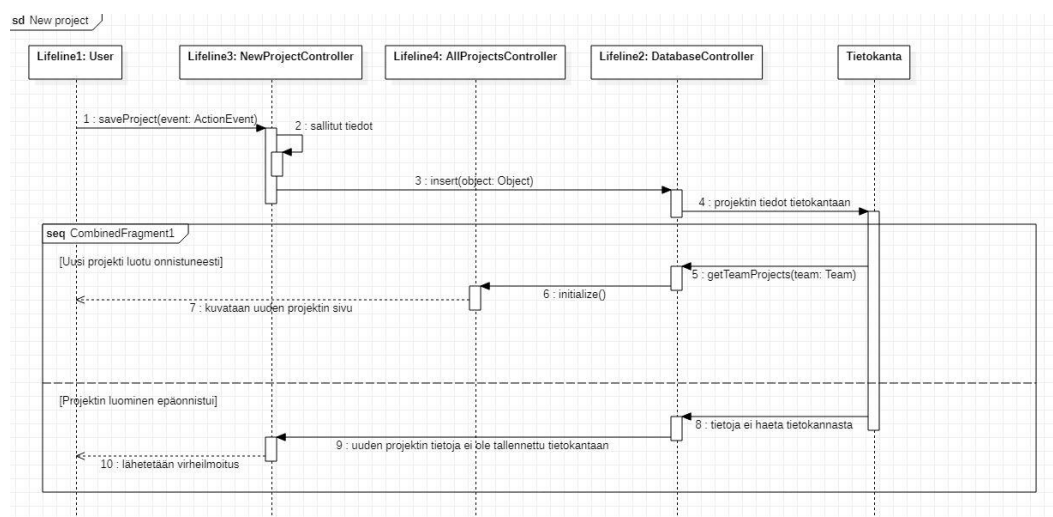
Sekvenssikaaviossa näkyy jäsenten lisääminen ryhmään ja poistaminen ryhmästä. Lisäksi jäsenten rooleja voi vaihtaa adminiksi tai useriksi, jota kaaviossa ei ole kuvattu. Rooleja vaihdetaan TeamInfoController -luokassa.



**Kuva 12.** Ryhmään lisätään jäsen, sekä sieltä poistetaan jäsen (perustettu luokkakaaviosta: kuva 6)

## 6.6 Projektin luominen

Uuden projektin luominen alkaa yläpalkista valitsemalla “Projects => New Project...”, ja se vaatii sisäänkirjautumista. Oinaisuuksiksi projekti ottaa projektin nimen, sekä kuvauksen projektista. Projektia luodessa myös valitaan minkä ryhmän alle se kuuluu. Käyttäjä voi valita vain ryhmän, jossa hän on ylläpitäjänä. Kun projekti on luotu, käyttäjälle näytetään uuden projektin pääkatsaus.



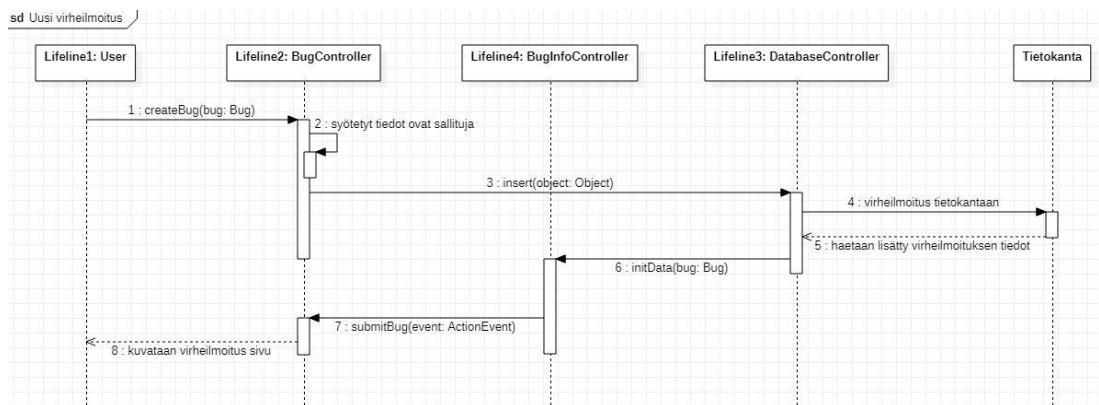
**Kuva 13.** Projektin luominen (perustettu luokkakaaviosta: kuva 6)

## 6.7 Projektin näkymä ja hallinta

Projektin hallinnasta vastaa AllProjectsController. Käyttäjän valinnasta riippuen kuvataan joko omat projektit tai kaikki projektit.

## 6.8 Uuden virheilmoituksen kirjaaminen

Projektin näkymästä voidaan luoda uusi virheilmoitus. Jotta käyttäjä voi luoda virheilmoituksen, hänen täytyy olla kirjautuneena sisään. Virheilmoitukseen käyttäjä kirjoittaa otsikon, sekä kuvauksen. Virheilmoituksesta myös tallennetaan minkä projektin alle se kuuluu, ilmoituksen tekijä, sekä päivämäärä. Jos virheilmoituksen luonnissa tapahtuu jokin virhe, siitä palautetaan virheilmoitus käyttäjälle. Kun käyttäjä on saanut virheilmoituksen luotua, käyttäjälle avataan uusi ikkuna, jossa on juuri luotu virheilmoitus.



**Kuva 14.** Uusi virheilmoitus (perustettu luokkakaaviosta: kuva 5)

## 6.9 Virheilmoituksen hallinta

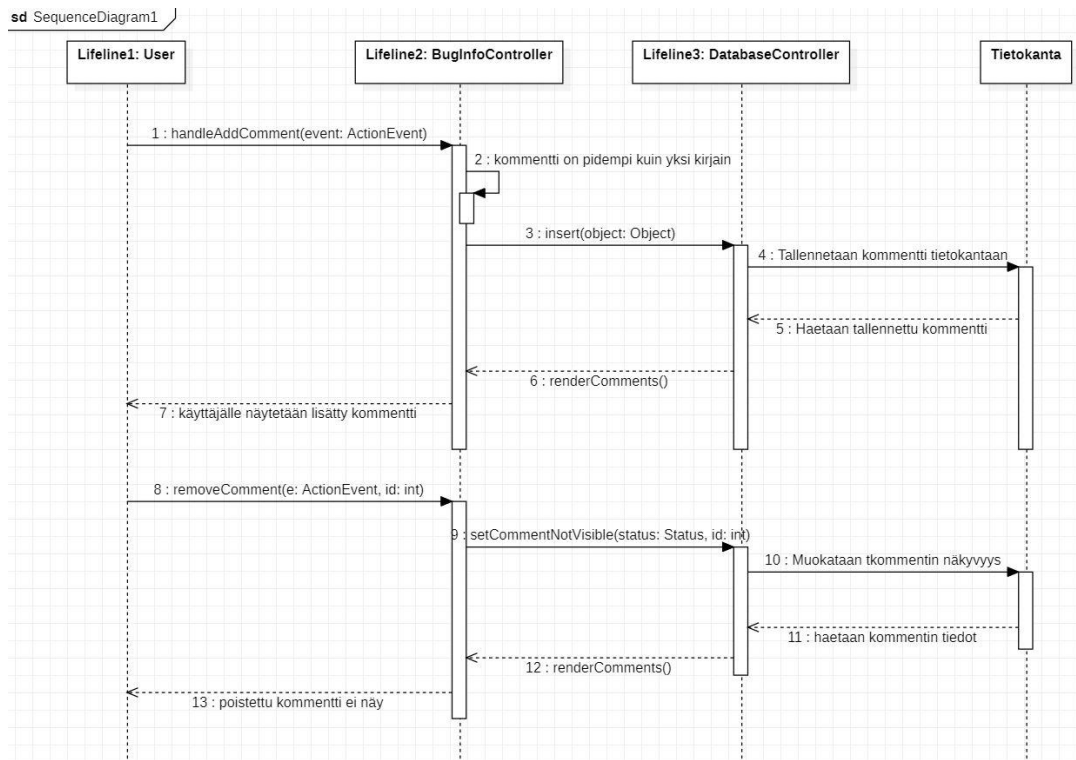
Projektille luodaan virheilmoituksia, kun tarvitaan apua. Kommentoijat esittävät omia ratkaisuja ja virheilmoituksen luoja voi merkitä lisätyt kommentit asiallisiksi, jos niistä tulee apua ja epäasiallisiksi, jos kommentti ei johda mihinkään.

Virheilmoitusten hallinnasta vastaa BugInfoController luokka. Käyttäjän valinnasta riippuen kuvataan joko avoimet virheilmoitukset, suljetut virheilmoitukset, aktiiviset virheilmoitukset tai uudet virheilmoitukset.

## 6.10 Kommentin lisääminen

Kirjautuneet käyttäjät voivat lisätä kommentteja sovelluksen virheilmoituksiin, joiden on tarkoitus edesauttaa ilmoittajaa virheen kanssa. Kommenttikenttä on virheilmoituksen loppuosassa. Kun kommentti lähetetään, se validoidaan ennen tallennusta tietokantaan. Jos tallennus onnistuu, käyttäjän näkymä päivitetään, jolloin käyttäjä näkee heti kommentin tallentuneen virheraporttiin.

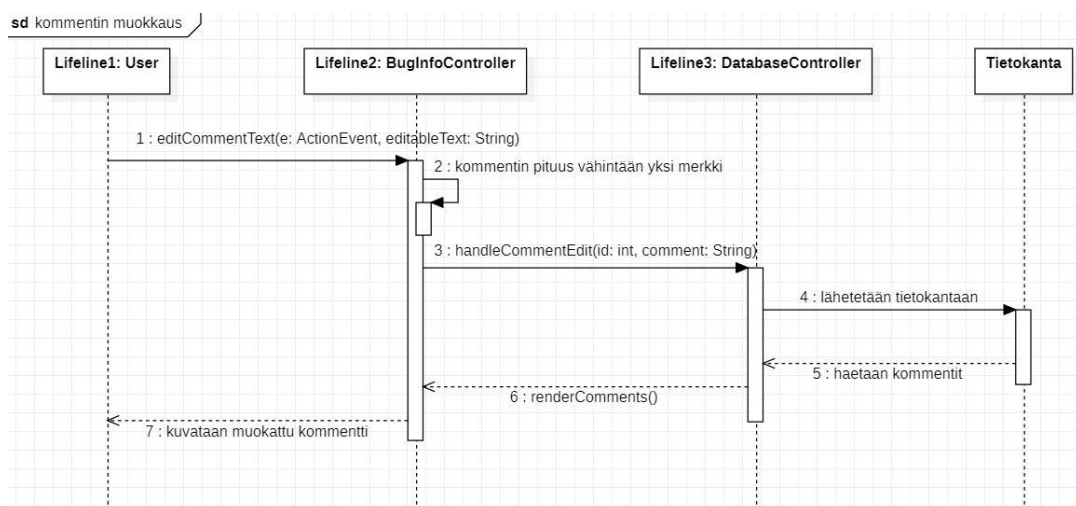




**Kuva 15.** Kommentin lisääminen ja poistaminen (perustettu luokkakaaviosta: kuva 5)

## 6.11 Kommentin muokkaaminen

Käyttäjän on oltava kirjautuneena sisään, jos hän haluaa kirjoittaa bugille kommentin, poistaa kommentin tai muokata kommenttia. Käyttäjä voi poistaa tai muokata vain omia kommentteja.



**Kuva 16.** Kommentin muokkaus (perustettu luokkakaaviosta kuva 5)

## 7 Kehitysprosessi ja kehitysvaiheen tekniikat

Tässä osiossa kuvataan kehitysprosessi sekä käytetyt kehitys- ja testausmenetelmät ja ympäristöt.

### 7.1 Kehitysprosessi

SuperBugTracker sovelluksen kehityksessä on erityisesti keskitytty käyttämään ketterää ohjelmistokehitystä. Projektin hallinnassa on käytetty Scrum menetelmää. Käytännössä siis kehitys on jaettu kahden viikon mittaisiin ”sprintteihin” ja jokainen ryhmäläinen on ollut vuorollaan Scrum Master. Projektin kehityksessä on myös pidetty Scrumiin kuuluvia Daily tapaamisia.

Tehtävien jakamisessa on käytetty Nektion Kanban boardia. Nektioniin on lisätty mitkä kukin ryhmäläinen on tehnyt ja tekee sprinttien aikana.

### 7.2 Kehitysmenetelmät

Sovelluksen lähdekoodin versionhallintaan on käytetty Git-ohjelmistoa, ja lähdekoodin tiedostot sijaitsevat Metropolian GitLab –palvelimella. Sillä projektissa on useita henkilöitä tekemässä muutoksia lähdekoodiin samanaikaisesti, on versionhallinnassa ”haaraumia”, johon kehittäjät kehittävät eri ominaisuuksia. Kun nämä ominaisuudet on kehitetty valmiiksi, ne yhdistetään päähaaraumaan, jolloin ne tulevat mukaan seuraavaan jakeluversioon.

### 7.3 Testaus

SuperBugTracker sovelluksessa on käytetty Junit5 testausympäristöä koodin automaatiotestien luomiseen. Kehityksen aikana kaikkiin käyttäjätarinoihin on ollut tavoitteena luoda automaatiotestejä, jotta kattavuus pysyy korkeana.

Itse testien suorittamisesta vastaa Jenkins, joka tarkastaa versionhallinnassa tapahtuvia muutoksia aina puolen tunnin välein, ja jos Jenkins havaitsee master

–rungossa uuden version, niin Jenkins ajaa automaatiotestit Mavenia käyttäen. Jenkins myös luo raportit testeistä, sekä koodikattavuudesta.

## 8 Käyttöohje

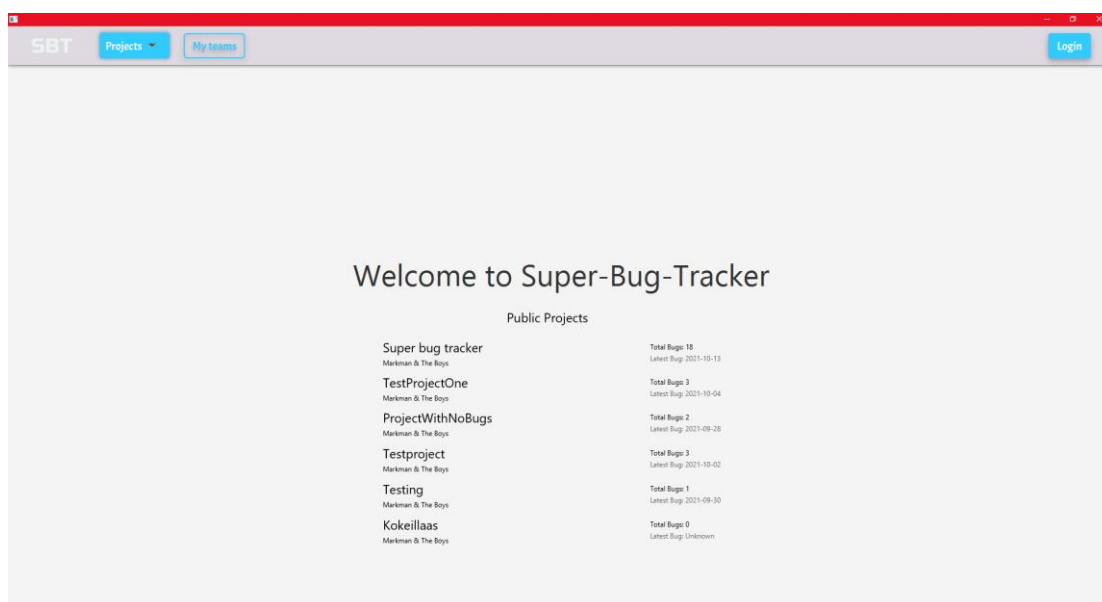
Tässä osassa kerrotaan, miten sovellus saadaan käyntiin ensimmäisen kerran. Tämä käyttöohje olettaa, että uudella käyttäjällä on asennettuna tarvittava Java-16 versio. Sovellus voidaan tällä hetkellä käynnistää vain IntelliJ editorilla.

Ennen sovelluksen käynnistämistä on käyttäjän oltava kirjautuneena Metropolian VPN-verkkoon. Tähän löytyvät ohjeet Metropolian omilta sivuilta hakemalla Googlestä esim. Metropolia vpn. Sovellus käynnistetään IntelliJ editorissa painamalla "MainApplication" tiedostosta "run" komentoa. "Run" löytyy luokan nimen vierestä vasemmalta.

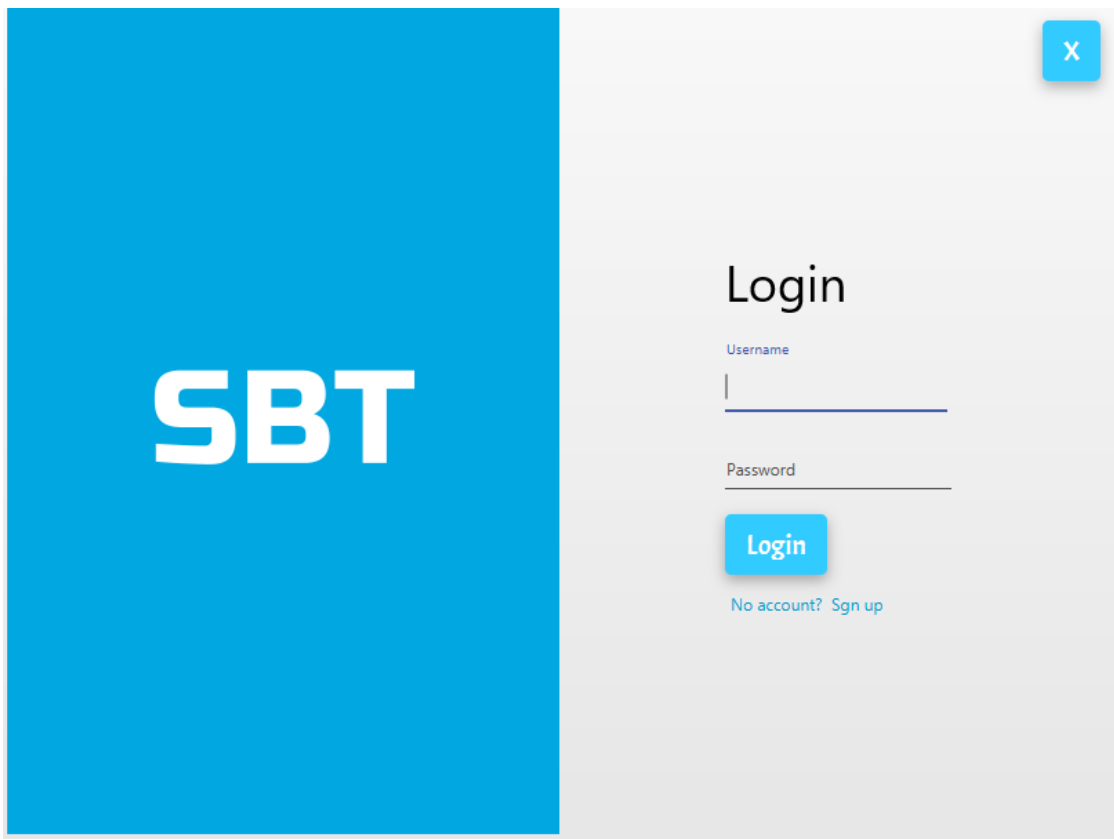
Sovellusta voi käyttää ilman käyttäjätunnuksia, mutta vain katsellakseen muita projekteja sekä virheilmoituksia.

### 8.1 Sisäänkirjautuminen/Rekisteröinti

Sovelluksen käynnistyessä aukeaa sovelluksen päänäköymä. Käyttäjän kannattaa aluksi luoda uusi profiili painamalla oikeasta yläkulmasta "Login" painiketta (kts. Kuva 17.) ja tämän jälkeen painaa sisäänkirjautumisen näkymässä "No account? Sign up" painiketta (kts. Kuva 18).

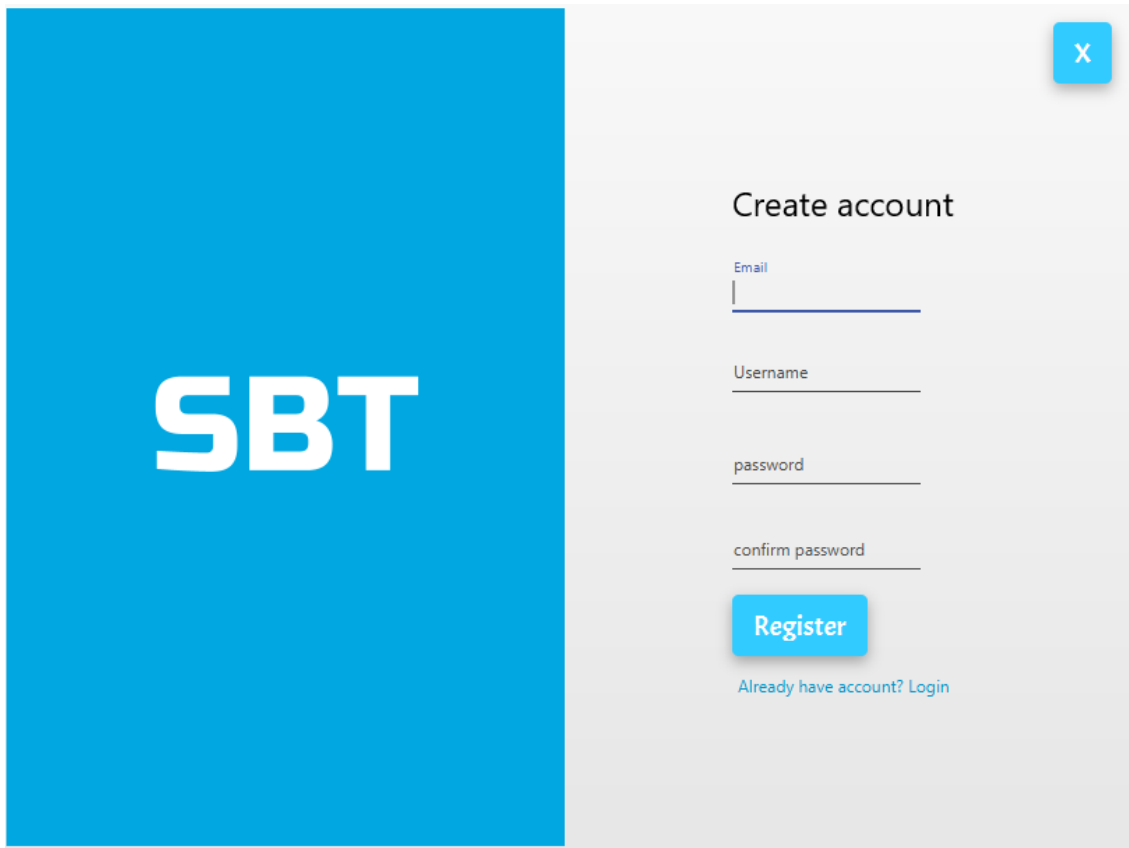


**Kuva 17.** Sovelluksen päänäköymä, kun ei ole kirjaututtu sisään.



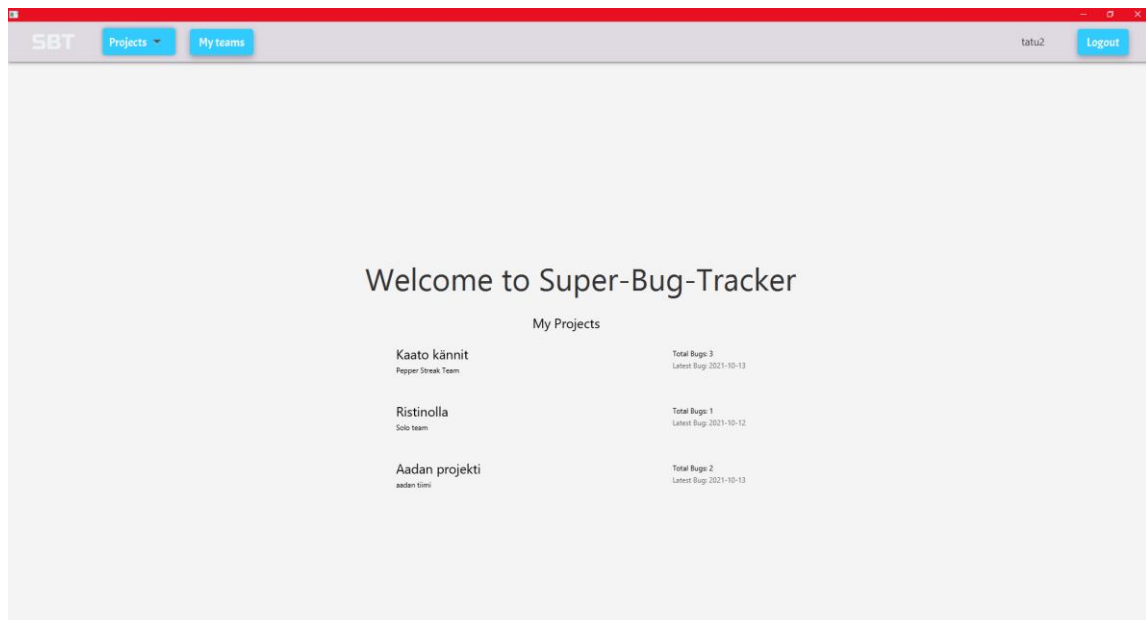
**Kuva 18.** Sisäänkirjautumisen näkymä.

Käyttäjän päästyä rekisteröitymisen näkymään, täytyy hänen täyttää kaikki kentät (kts. Kuva 19).

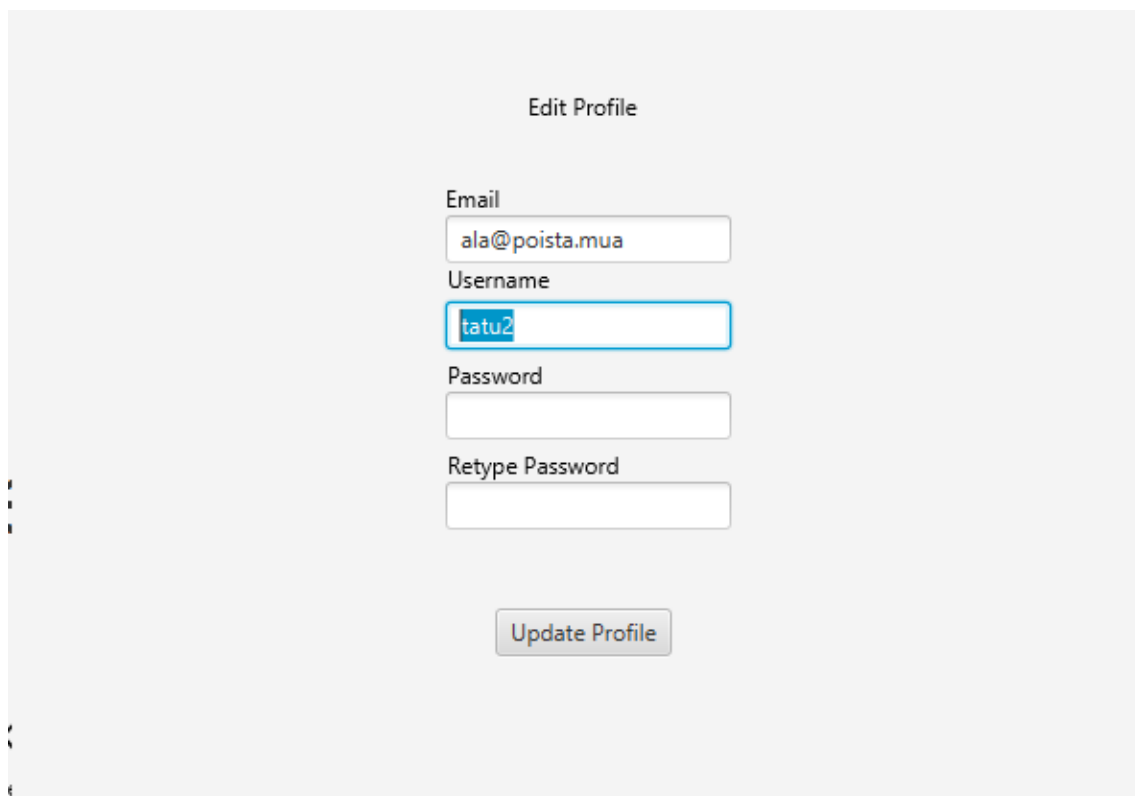
The image shows a registration form for a service called SBT. On the left, there is a large blue rectangle with the white text "SBT" in the center. To the right of this rectangle is a light gray area containing the registration form. In the top right corner of the gray area is a small blue square with a white "X" icon. The form is titled "Create account" in bold black text. Below the title are four input fields: "Email", "Username", "password", and "confirm password". Each field has a small label above it and a horizontal line for text entry. Below the "password" field is a blue button with the white text "Register". At the bottom of the form, there is a link that says "Already have account? Login" in blue text.

**Kuva 19.** Rekisteröitymisen näkymä.

Onnistuneen sisäänkirjautumisen huomaa oikeasta ylänurkasta, jossa käyttäjätunnus näytetään ja "Login" painikkeen tilalle on tullut "Logout" painike (kts. Kuva 20). Painamalla käyttäjätunnusta pääsee myös käyttäjän tietojen muokkauksen näkymään (kts. Kuva 21) ja "Logout" painikkeesta luonnollisesti kirjautetaan ulos sovelluksesta.



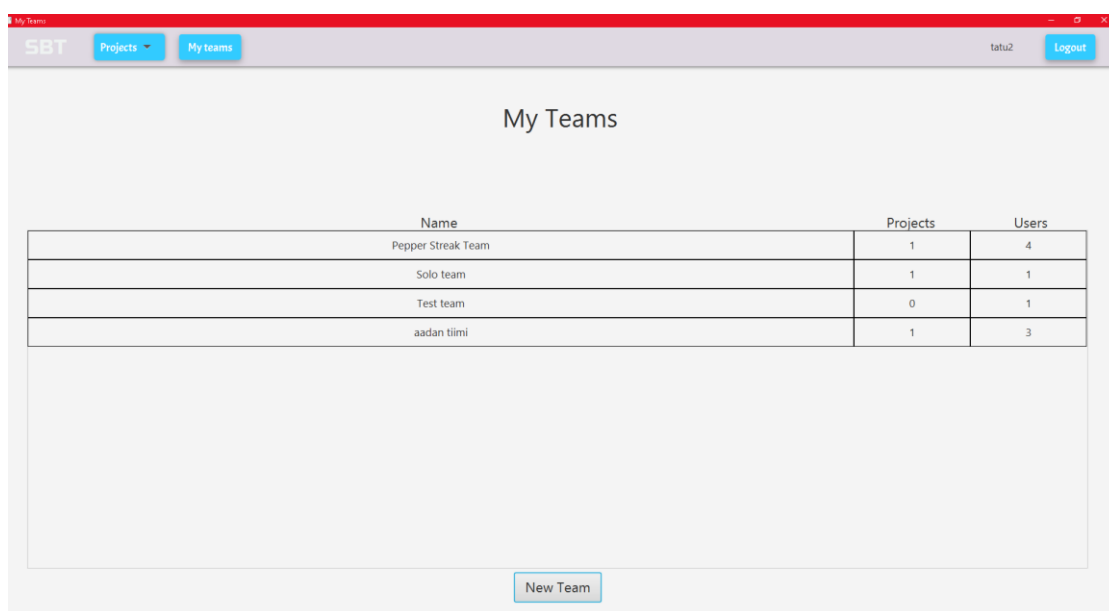
**Kuva 20.** Sisään kirjautuneen käyttäjän päänäkö.



**Kuva 21.** Profiilin muokkauksen näkö. HUOM! Keskeneräinen toteutus.

## 8.2 Tiimin luonti ja tiimin käyttäjien hallinta

Uudella käyttäjällä ei ole virheilmoituksia, projekteja eikä tiimejä vielä olemassa ja nämä täytyy luoda järjestyksessä. Aluksi käyttäjän täytyy luoda tiimi. Tiimin luontiin pääsee painamalla päänäköymässä “My teams” painiketta (kts. Kuva 20), joka vie käyttäjän omien tiimiensä listauksen näkymään ja jossa sijaitsee “New Team” painike (kts. Kuva 22). Tiimi voidaan myös luoda “Projects” pudotusvalikon kautta siirtymällä “New Project...” kohtaan, mutta se vie vain takaisin kuvan 22 näkymään.



Name	Projects	Users
Pepper Streak Team	1	4
Solo team	1	1
Test team	0	1
aadan tiimi	1	3

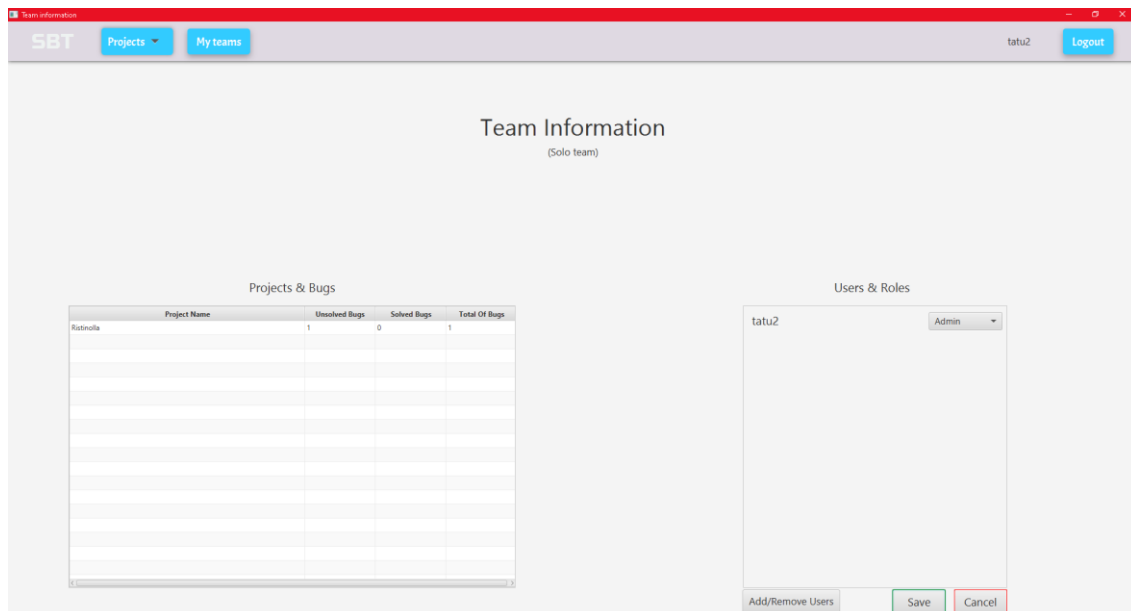
**Kuva 22.** Omien tiimien näkymä. Näkymässä näkyy omat tiimit, tiimien projektien määrät, tiimien jäsenmäärät sekä painike uuden tiimin luonnille.

Käyttäjän painettuaan “New Team” painiketta, avautuu käyttäjälle uuden tiimin luonnin ikkuna (kts. Kuva 23). Tähän käyttäjän täytyy antaa tiimille vain tiimin nimi. Käyttäjien lisääminen on vapaaehtoista. Käyttäjiä voi lisätä tiimiin valitsemalla pudotusvalikosta käyttäjien tunnuksia. Käyttäjät voidaan myös poistaa tiimistä painamalla käyttäjätunnuksen vieressä olevaa “Remove” painiketta, jos sattuu vahinko.



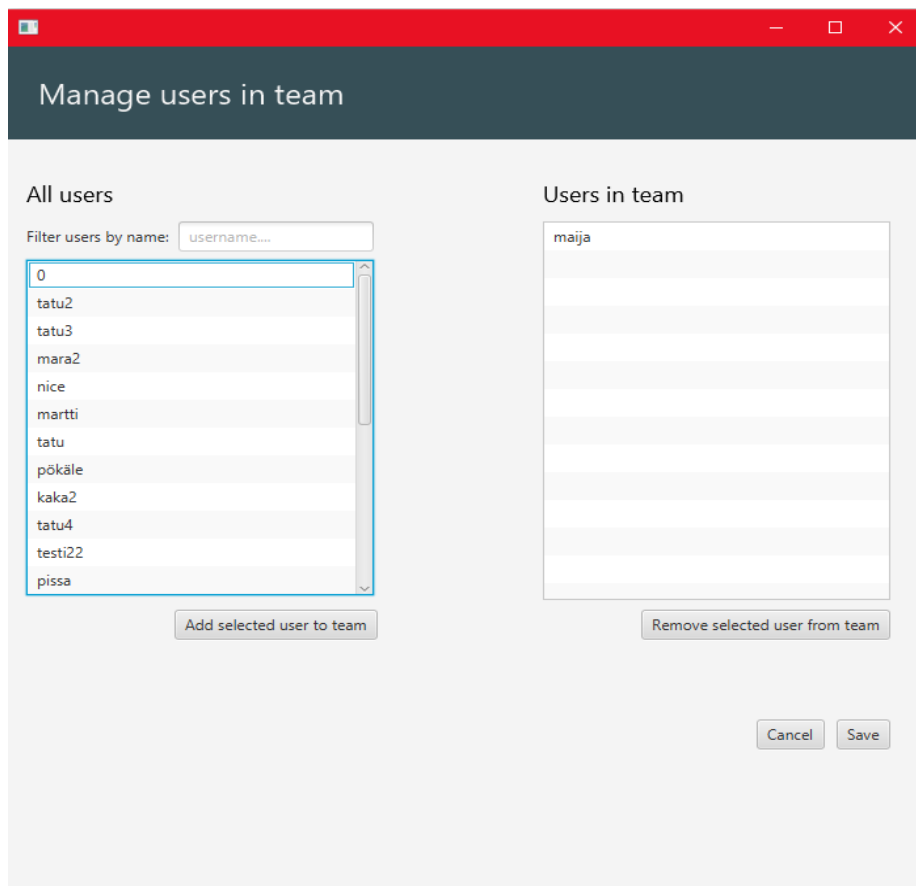
**Kuva 23.** Uuden tiimin luonnin ikkuna.

Kun tiimi on olemassa, voidaan sen käyttäjiä sekä niiden rooleja hallita painamalla tiimin nimeä (kts. Kuva 22). Tästä avautuu uusi näkymä, jossa esitellään tiimin projektit, virheilmoitusten lukumäärät sekä käyttäjät ja niiden roolit tiimissä. Käyttäjien rooleja voidaan vaihdella pudotusvalikosta valitsemalla rooli ja lopuksi painaa "Save" painiketta (kts Kuva 24). Painike "Cancel" peruu kaikki roolien muutokset ja päivittää näkymän.



**Kuva 24.** Tiimin tietojen näkymä.

Tiimiin voidaan lisätä käyttäjiä sekä poistaa niitä siirtymällä käyttäjien hallintaan painikkeesta “Add/Remove Users” (kts. Kuva 24). Tässä näkymässä (kts. Kuva 25) voidaan lisätä “All users” laatikosta käyttäjiä ensin valitsemalla jonkin käyttäjätunnuksen ja sitten painamalla “Add selected user to team”, joka siirtää käyttäjän “Users in team” listaukseen. Käyttäjiä voidaan myös poistaa samalla tyylillä, eli valitsemalla “Users in team” listauksesta käyttäjä ja painamalla “Remove selected user from team”. Lopuksi täytyy painaa painiketta “Save”, jos muutokset halutaan tallentaa. Muutoksien kumoamiseen voidaan käyttää painiketta “Cancel”, joka sulkee ikkunan ja peruuttaa muutokset.



**Kuva 25.** Käyttäjien hallinta tiimin sisällä.

### 8.3 Projektin luonti ja projektien näkymät

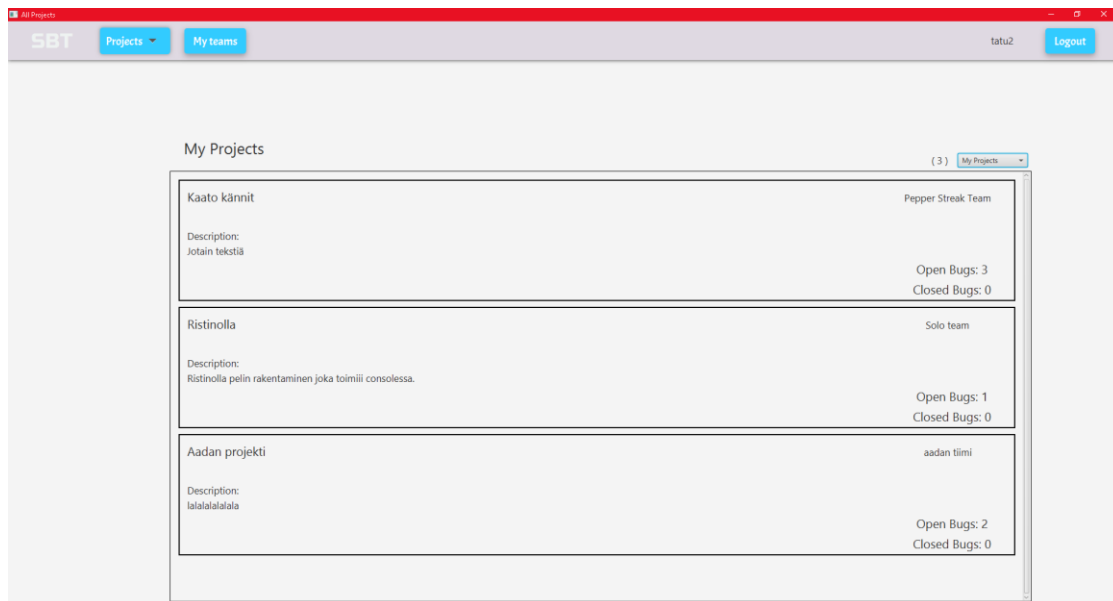
Uuden projektin voi luoda painamalla pudotusvalikkoa "Projects" ja valitsemalla "New project..." (kts. Kuva 20). Projektille täytyy antaa nimi sekä valita mihin tiimiin projekti kuuluu, mutta projektin kuvaus on vapaaehtoinen (kts. Kuva 26). Jos käyttäjä ei kuulu mihinkään tiimiin niin käyttäjälle ilmoitetaan, että tiimi täytyy luoda ja näkymään tulee nappi, joka siirtää käyttäjän tiimien listaus näkymään (kts. Kuva 22), josta käyttäjä voi luoda uuden tiimin. Onnistuneen projektin luonnin jälkeen siirtyy näkymä projektin virheilmoitusten näkymään (kts. Kuva 27).

**Kuva 26.** Uuden projektin luonnin näkymä.

Kun projektia painetaan, joko päänäköymästä (kts. Kuva 20), projekti luodaan, avataan “My projects” näköymästä (kts. Kuva 28) tai “All projects” näköymästä (kts. Kuva 29), niin siirrytään projektin virheilmoitusten näköymään (kts. Kuva 27), jossa siis esitellään kaikki projektin sisäiset virheilmoitukset. Kaikkien projektien ja omien projektien näkymä on hyvin samankaltainen kuin projektin virheilmoitusten näkymä, mutta ne erottavat siitä, että filterit ovat erilaisia, otsikot eivät ole samankaltaisia (yleensä virheilmoituksen otsikko ei olisi projektin nimen kaltainen) sekä muutamasta napista “New Bug report” ja “Information”.

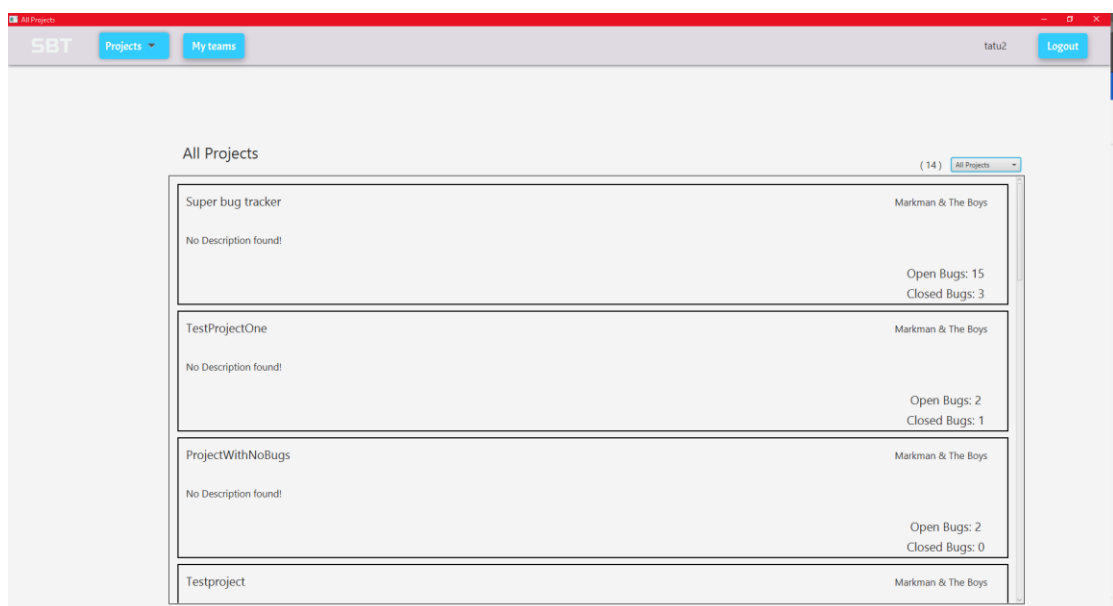
**Kuva 27.** Projektin virheilmoitusten listauksen näkymä.

Projektien näkymässä voidaan erotella filtterin avulla (kuvassa pudotusvalikko, jossa lukee “My Projects”) omat projektit ja kaikki projektit, jotka ovat julkisesti saatavilla.



**Kuva 28.** Omien projektien näkymä.

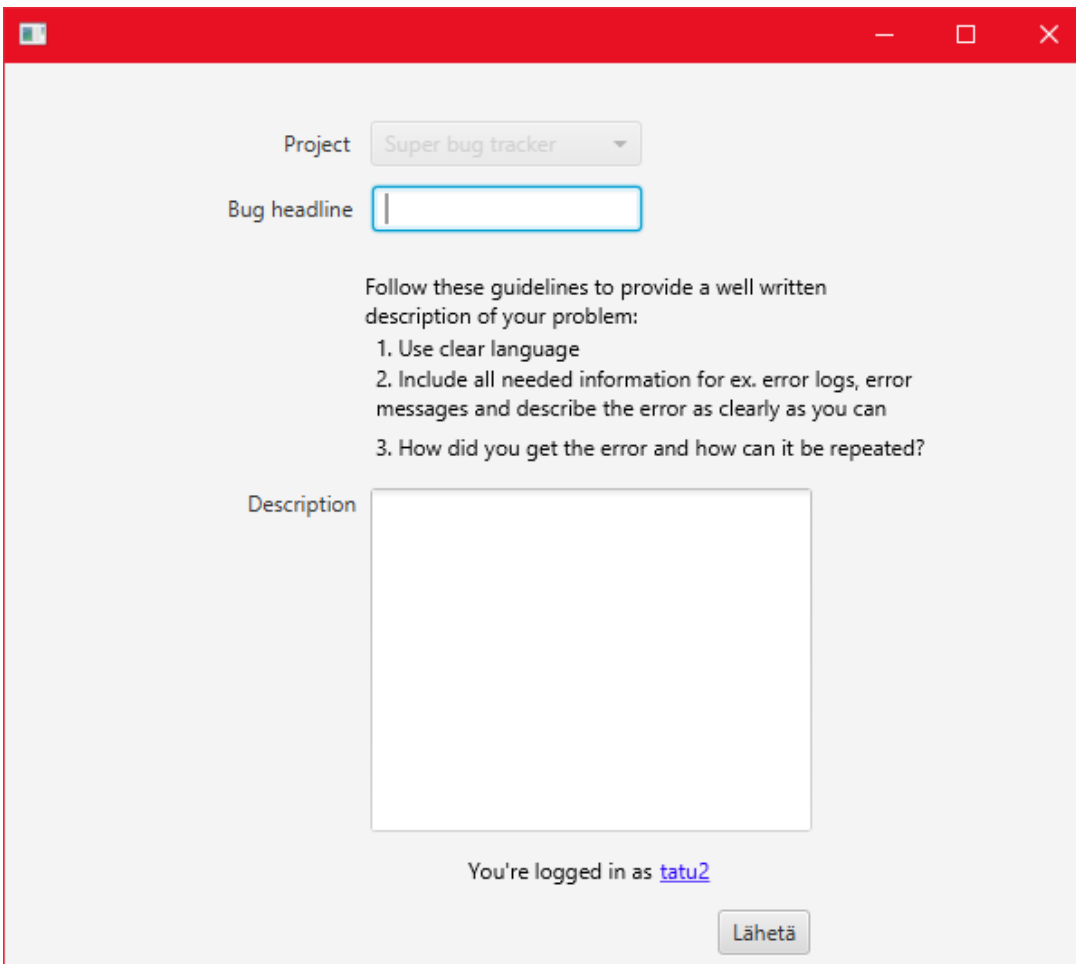
Kaikkien projektien näkymä on muuten samankaltainen kuin omien projektien, mutta otsikko ja pudotusvalikon tekstit muuttuvat.



**Kuva 29.** Kaikkien projektien näkymä.

## 8.4 Virheilmoituksen luonti ja niihin kommentointi

Uuden virheilmoituksen voi luoda projektin virheilmoitusten näkymästä painamalla painiketta “New bug report” (kts. Kuva 27). Tästä aukeaa uuden virheilmoituksen ikkuna (kts. Kuva 30). Virheilmoitukselle tulee automaattisesti projekti, johon se luodaan. Virheilmoitukselle täytyy antaa otsikko sekä kuvaus. Kuvausta on opastettu hieman, jotta käyttäjä ei julkaisisi liian epäselvää ilmoitusta. Painamalla painiketta “Send” virheilmoitus lisätään projektin alle kuuluviin virheilmoituksiin.



The screenshot shows a web application window titled "Super bug tracker" with a red header bar. The form contains the following elements:

- A "Project" dropdown menu with "Super bug tracker" selected.
- A "Bug headline" text input field.
- Guidelines for writing a description:

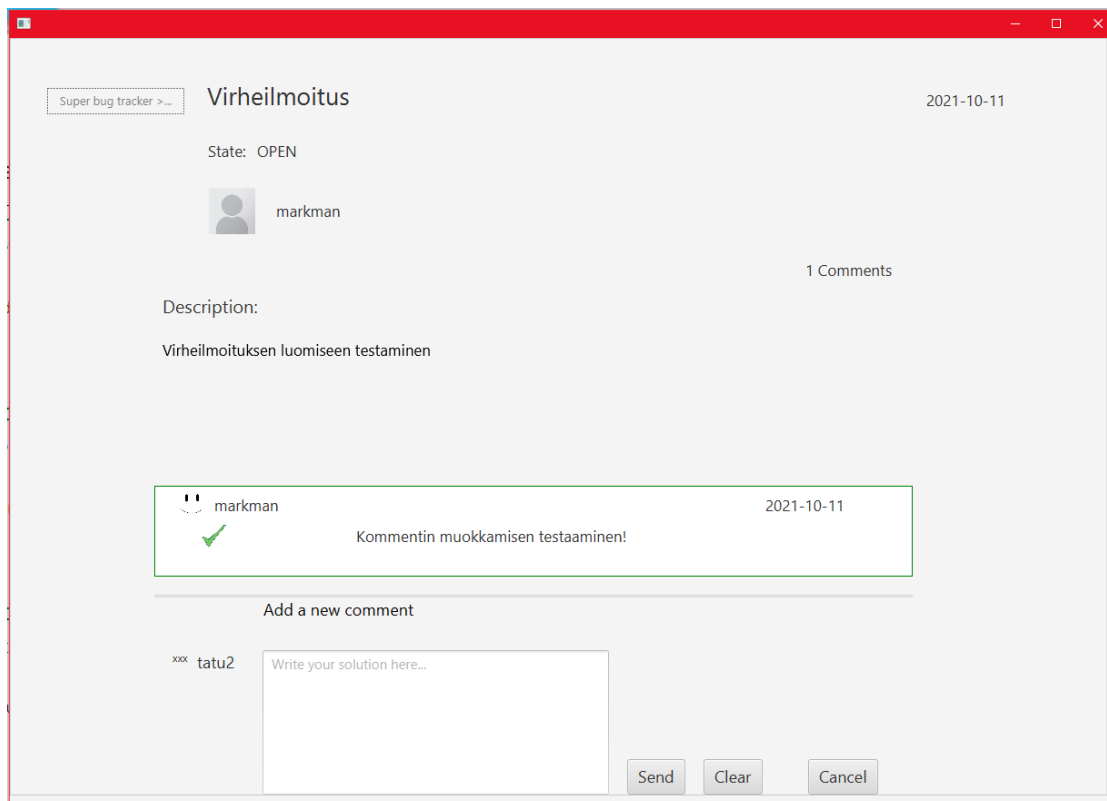
Follow these guidelines to provide a well written description of your problem:

  1. Use clear language
  2. Include all needed information for ex. error logs, error messages and describe the error as clearly as you can
  3. How did you get the error and how can it be repeated?
- A "Description" text area.
- A login status message: "You're logged in as [tatu2](#)".
- A "Lähetä" (Send) button.

**Kuva 30.** Virheilmoituksen luonnin ikkuna.

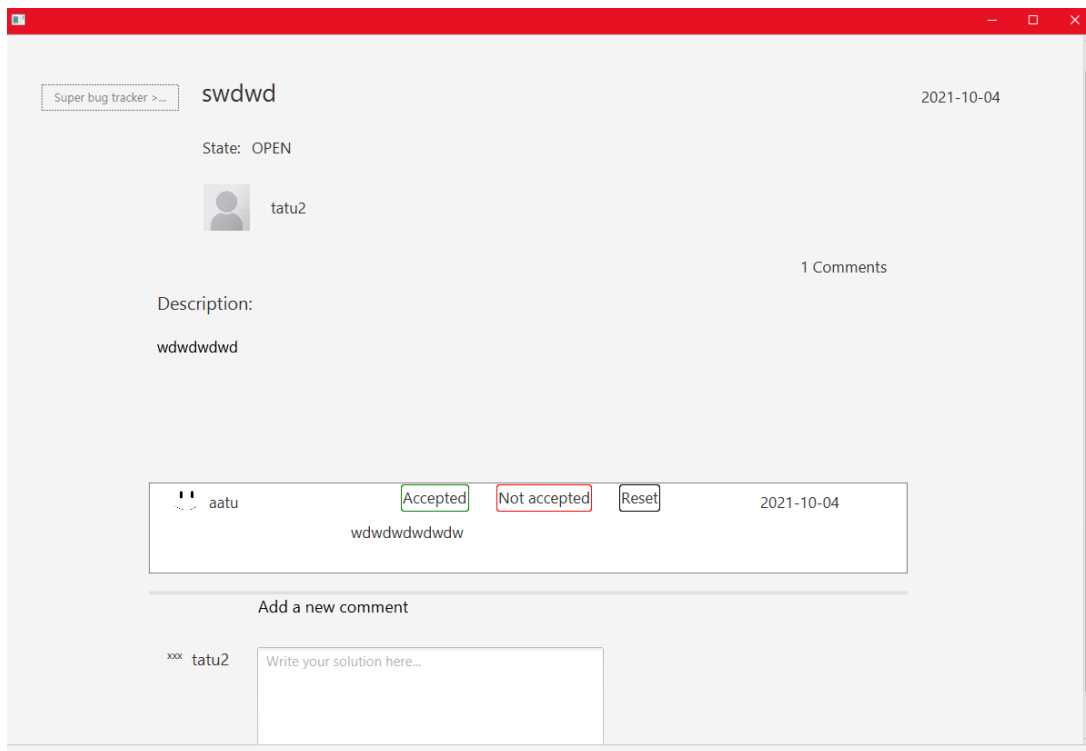
Virheilmoitusten tietoja voidaan katsella painamalla virheilmoituksen aluetta (kts. Kuva 27, alueet ovat rajattu, joko vihreällä tai punaisella riippuen ilmoituksen tilasta). Tämä vie näkymään, jossa on kaikki ilmoituksen oleelliset tiedot sekä kommenttien listaus (kts. Kuva 31). Käyttäjä voi lisätä kommentteja

ilmoitukseen kirjoittamalla tekstikenttään ratkaisunsa ja painamalla “Send” painiketta. Painike “Clear” tyhjentää koko tekstikentän ja painike “Cancel” sulkee ikkunan. Näkymä avautuu omaksi ikkunaksi, jotta sitä voidaan liikutella ja pitää auki, kun muuten sovellusta käytetään. Ikkunan voi sulkea painamalla otsikon vieressä olevaa projektin nimeä tai “X” painiketta nurkasta.

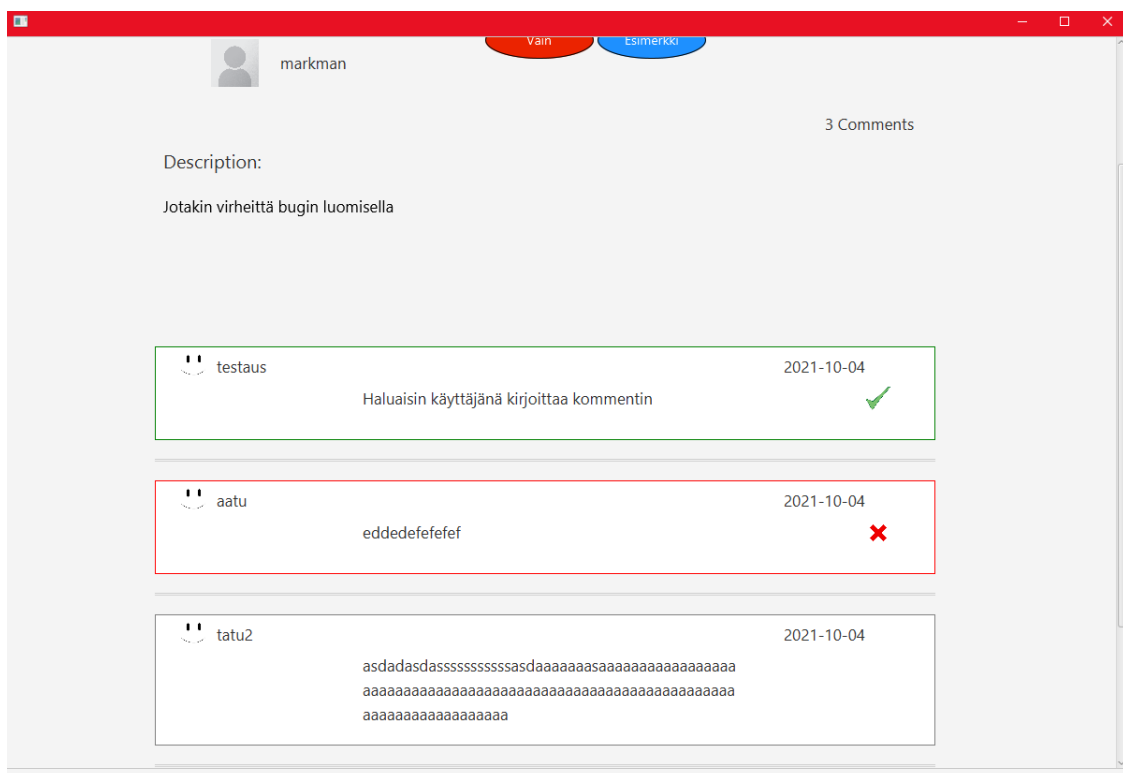


**Kuva 31.** Virheilmoituksen tietojen näkymä.

Virheilmoituksen tekijä voi määrittää kommenttien laadun valitsemalla kommentin kohdalla olevista painikkeista joko “Accepted”, “Not accepted” tai nollaamalla kommentin laadun painamalla “Reset” (kts. Kuva 32). Näin syntyy kommentille huomattava merkintä siitä onko kommentti oleellinen (kts. Kuva 33).



**Kuva 32.** Kommentin merkitä. Kommentin voi merkitä hyväksytyksi tai ei hyväksytyksi (Vain ilmoituksen tekijän oikeus).

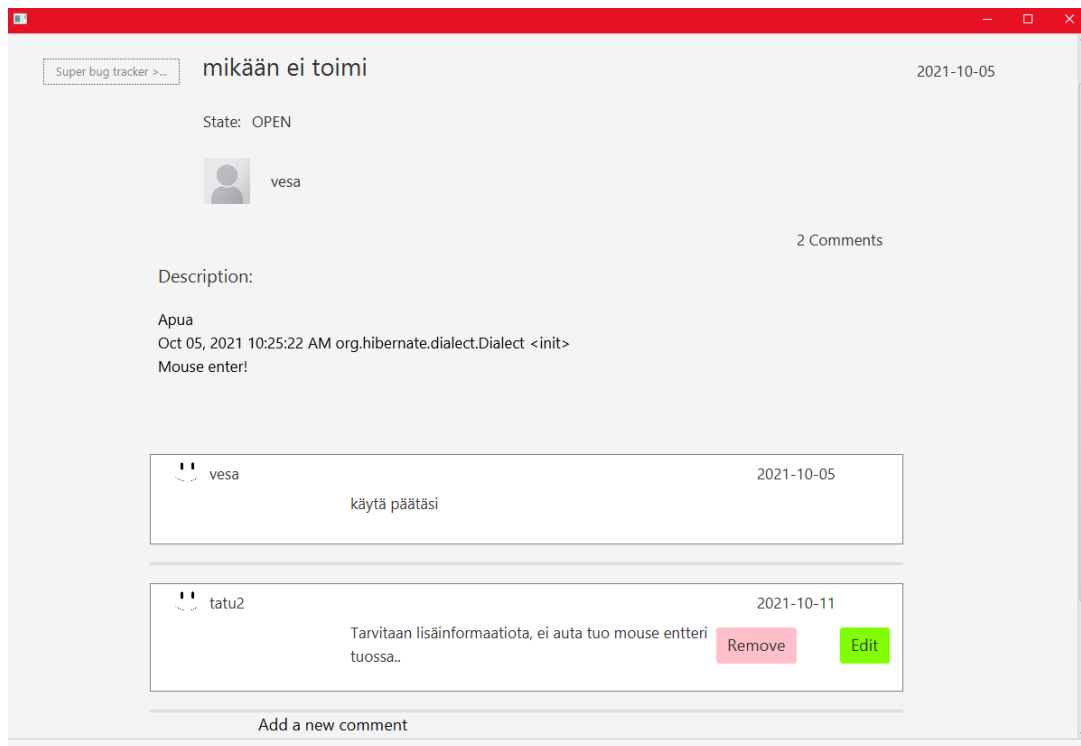


**Kuva 33.** Kommenttien oleellisuuden merkinnät. Ristillä tarkoitetaan, että



kommentti ei ole oleellinen tai on huono ratkaisu ja “tehty” merkinnällä ilmoitetaan, että kommentti on ollut hyödyllinen.

Kommentin laatija voi myös muokata tekemäänsä kommenttia tai poistaa sen kokonaan (kts. Kuva 34).



**Kuva 34.** Kommentin muokkaus sekä poistaminen.

## 9 Yhteenveto

SuperBugTracker sovelluksen kehitys on hyvässä vauhdissa ja sovellus alkaa näyttää jo valmiilta tuotteelta. Tekemistä kuitenkin riittää vielä kaikkien ominaisuuksien tyylyttelyn suhteen ja myös paljon bugeja riittää korjattavaksi.

Tavoitteisiin on päästy erinomaisesti. Ryhmällä oli tavoitteena saada ensimmäisen periodin aikana toimiva tuote, josta löytyy määritetyt perusominaisuudet, jota virheidenhallintasovelluksessa voi olla, ja ryhmä sai kaikki nämä toteutettua.

Jatkokehitys pohjautuu paljon aiemmin kirjoitettuihin käyttäjätarinoihin. Sovelluksen jatkokehityksiin kuuluu myös OTP2-kurssin kansainvälistämisen ominaisuudet.

Jatkokehityksen aiheita:

- Bugeille tagien antaminen sekä liitteiden lisäämisen ominaisuus.
- Projektien, bugien, tiimien ja käyttäjän poisto mahdollisuus.
- Käyttäjän salasanan muokkaukseen vanhan salasanan lisäys, sekä salasanan palautus mahdollisuus.
- "Tooltips" tyylisiä ohjeistuksia sovelluksen eri kohtiin, jotta sovelluksen käyttö helpottuisi entistä enemmän.