Point Fight Scrum -tiimi

Ronja Kaskela, Toni Rehnman

Scrum ja ohjelmistotuotannon menetelmät -kurssi  
K2020, Careeria

SUUNNITTELUKUVASTO

Point Fight -projekti

Sisällysluettelo

[1 Järjestelmän tekninen arkkitehtuurikuvaus 2](#_Toc40866147)

[2 Käyttöliittymä-/sivustokaaviot 3](#_Toc40866148)

[3 Tietohakemisto 4](#_Toc40866149)

[4 Tietokannan ER-malli 5](#_Toc40866150)

[5 Tietokannan taulujen luontilauseet 6](#_Toc40866151)

[5.1 Hahmot 6](#_Toc40866152)

[5.2 Käyttäjät 6](#_Toc40866153)

[5.3 Tehtävät 6](#_Toc40866154)

[5.4 ValitutTehtävät 7](#_Toc40866155)

[6 Luokkakaaviot 8](#_Toc40866156)

[6.1 Luokat 9](#_Toc40866157)

[6.2 Metodit 9](#_Toc40866158)

[7 CRUD-taulukko 10](#_Toc40866159)

[8 Dokumentin versionhallinta 11](#_Toc40866160)

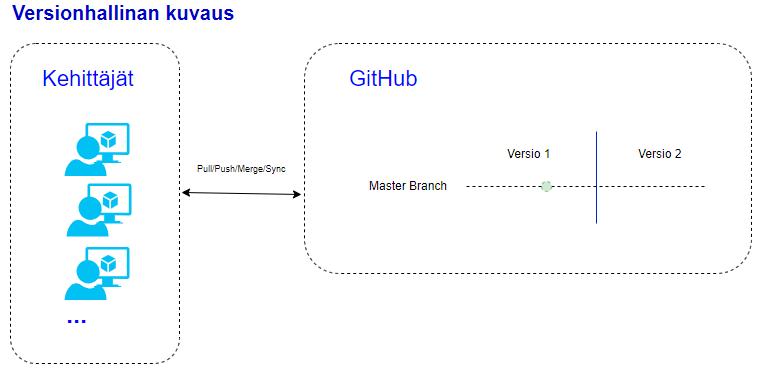
# Järjestelmän tekninen arkkitehtuurikuvaus

**Yleistä**

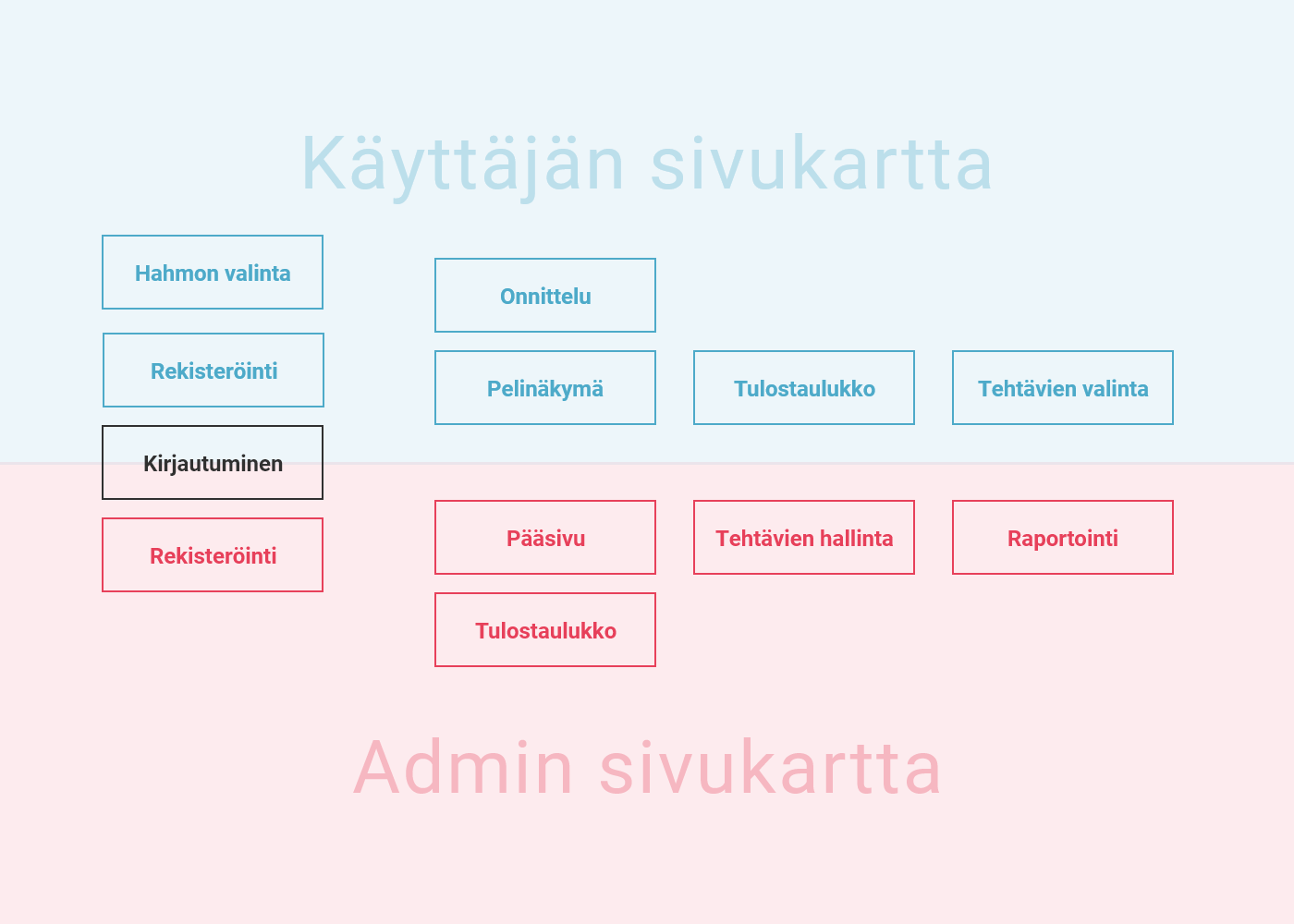
Ohjelmistokehityksessä työvälineinä käytetään Visual Studio Codea, Adobe Illustratoria, Adobe Photoshopia ja Adobe After Effectsiä. Ohjelmointikielenä käytetään JavaScriptiä ja React.js-kirjastoa. Sovelluksen tietokantana käytetään Amazon Web Services(AWS) SQL-Server -palvelua. Sovelluskehitys tehdään lokaalisti kehittäjien koneilla ja valmis sovellus julkaistaan missä?.

**Versionhallinta**

Versionhallinta toteutetaan Git-versionhallintaohjelmistolla, josta koodi siirretään GitHub-versionhallintapalveluun. Tätä varten kehittäjille on luotu tilit Github-palveluun. Kehittäjät tekevät kehitystyötä paikallisessa GIT-repositoriossa, josta se siirretään GitHub versionhallintapalveluun.

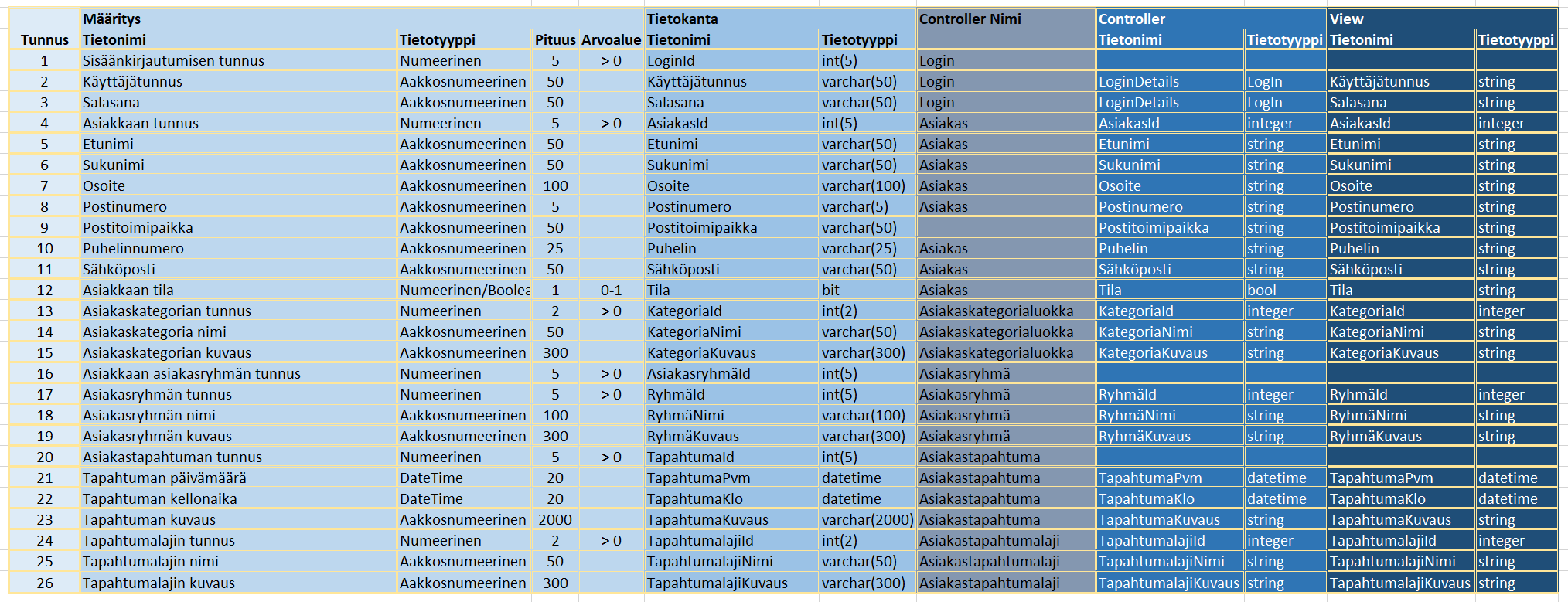


# Käyttöliittymä-/sivustokaaviot



# Tietohakemisto

Tietohakemisto esitetään tässä kuvastossa kuvana. Sen ylläpito projektin aikana tapahtuu kuitenkin alkuperäisessä Excel-taulussa (tietohakemisto.xlsx), joka on tämän kuvaston liitteenä.



# Tietokannan ER-malli

Tietokannan ER-malli esitetään tässä kuvana. Sen ylläpito tapahtuu tarvittaessa Microsoft SQL Server Management Studiossa. (tietohakemisto.xlsx), joka on tämän kuvaston liitteenä)

Kuva, joka sisältää kohteen näyttökuva

Kuvaus luotu automaattisesti

# Tietokannan taulujen luontilauseet

## Hahmot

CREATE TABLE Hahmot(

hahmo\_id int IDENTITY (10,1) NOT NULL,

nimi nvarchar (30) NOT NULL,

animaatio1 nvarchar(100),

animaatio2 nvarchar(100),

animaatio3 nvarchar(100),

PRIMARY KEY (hahmo\_id),

)

GO

## Käyttäjät

CREATE TABLE Käyttäjät(

käyttäjä\_id int IDENTITY (5,1) NOT NULL,

hahmo\_id int NOT NULL,

etunimi nvarchar (30) NOT NULL,

sukunimi nvarchar(30) NOT NULL,

salasana nvarchar(30) NOT NULL,

sähkposti nvarchar(50) NOT NULL,

rekisteröintikoodi int NOT NULL,

uusipäivä BIT,

yritys nvarchar(100),

admin BIT,

PRIMARY KEY (käyttäjä\_id),

FOREIGN KEY (hahmo\_id) REFERENCES Hahmot(hahmo\_id)

)

GO

## Tehtävät

CREATE TABLE Tehtävät(

tehtävä\_id int IDENTITY (10,1) NOT NULL,

nimi nvarchar (30) NOT NULL,

pistemäärä int,

tavoitemäärä int,

päivämäärä DATE,

pakollinen BIT,

yrityskoodi int NOT NULL,

PRIMARY KEY (tehtävä\_id)

)

GO

## ValitutTehtävät

CREATE TABLE ValitutTehtävät(

valitutteht\_id int IDENTITY (10,1) NOT NULL,

käyttäjä\_id int NOT NULL,

tehtävä\_id int NOT NULL,

pisteet int,

viimeinenmuokkaus DATE,

PRIMARY KEY (valitutteht\_id),

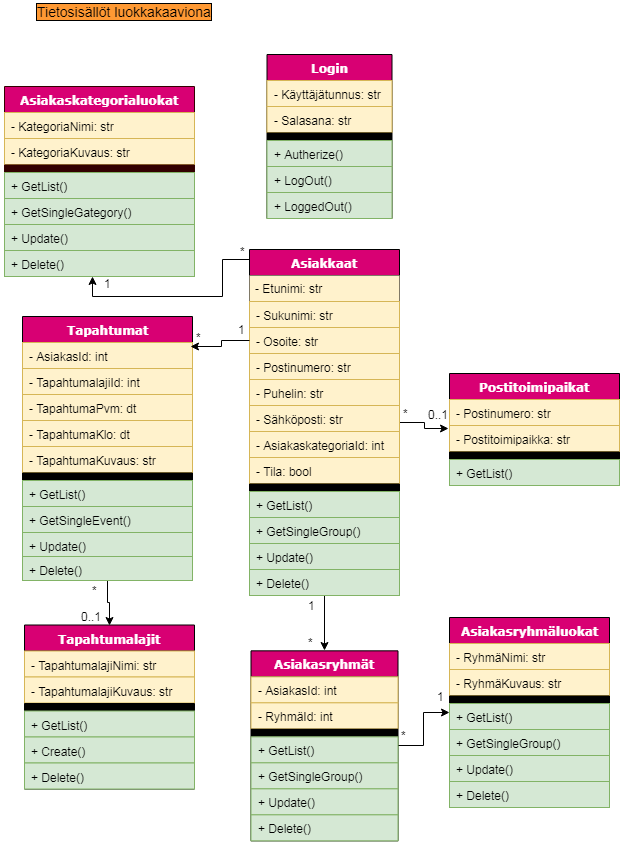
FOREIGN KEY (käyttäjä\_id) REFERENCES Käyttäjät(käyttäjä\_id),

FOREIGN KEY (tehtävä\_id) REFERENCES Tehtävät(tehtävä\_id)

)

GO

# Luokkakaaviot



Sovelluksen toiminnot muistuttavat toisiaan eri tietokannan luokissa.

## Luokat

**Hahmot**

* Pelattavat hahmot
* Hahmojen animaatiot

**Käyttäjät**

* Käyttäjien ja adminin tiedot
* Adminin erotus käyttäjästä

**Tehtävät**

* Tehtävien luonti
* Tavoitteiden ja pisteiden tallennus

**ValitutTehtävät**

* Pelaajan valitut tehtävät
* Pelaajan saavutetut pisteet

## Metodit

**Autherize() (Login)**

* Sisäänkirjautuminen järjestelmään (käyttäjätunnus ja salasana)
* Istunnon keston hallinta

**LogOut()**

* Uloskirjautuminen järjestelmästä

**LoggedOut()**

* Järjestelmä ilmoittaa uloskirjautumisesta
* Palauttaa sisäänkirjautumisikkunan

**Create() (vain Tapahtumalajit)**

* Luodaan uusi tapahtumalaji tai päivitetään olemassa olevaa

**GetList()**

* Haetaan tietokannasta kyseisen taulun kaikki tiedot

**GetSingleGroup(), GetSingleEvent(), GetSingleGategory()**

* Haetaan käyttöliittymässä valitun rivin tiedot tietokannasta

**Update()**

* Päivitetään käyttöliittymässä valitun rivin tietoja tietokantaan

**Delete()**

* Poistetaan käyttöliittymässä valitun rivin tiedot tietokannasta

# CRUD-taulukko

* Eri tietojen käyttöyhteenveto
* Käyttötapaus vrs. luokka
* C=Create, R=Read,U=Update,D=Delete

Jatkokehitys

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Luokka Käyttötapaus | Asiakas | Asiakas-kategoria | Asiakas-ryhmä | Asiakas-tapahtuma | Raportti | Sähköposti |
| Asiakastiedot  ylläpito | C, R, U, D | R, U | C, R, U, D | C, R, U, D |  |  |
| Asiakaskategoria  ylläpito |  | C, R, U, D |  |  |  |  |
| Asiakasryhmä  ylläpito |  |  | C, R, U, D |  |  |  |
| Asiakastapahtuma  ylläpito |  |  |  | C, R, U, D |  |  |
| Raportin ajaminen | R | R | R | R | C | Jatkokehitys |
| Sähköpostin lähettäminen | R |  |  |  |  | C |

# Dokumentin versionhallinta

|  |  |  |  |
| --- | --- | --- | --- |
| Versio | Pvm | Muutos | Tekijä |
| 1 | 11.5.2020 | Dokumentin luonti, tietokannan er-malli | Ronja Kaskela |
| 1.2 | 20.5.2020 | Lisätty tietokannan taulujen luontilauseet, sivukartta, luokat | Ronja Kaskela |
| 1.3 |  |  |  |