CRM Scrum-tiimi

Mika Purmonen, Sanna Ketomäki, Simo Sirén, Timo Tuiskunen, Tuula Yoshinari

Scrum ja ohjelmistotuotannon menetelmät -kurssi  
K2019, Careeria

SUUNNITTELUKUVASTO

CRM projekti

Sisällysluettelo

[1 Järjestelmän tekninen arkkitehtuurikuvaus 2](#_Toc11252552)

[2 Käyttöliittymä-/sivustokaaviot 3](#_Toc11252553)

[3 Tietohakemisto 3](#_Toc11252554)

[4 Tietokannan ER-malli 4](#_Toc11252555)

[5 Tietokannan taulujen luontilauseet 5](#_Toc11252556)

[5.1 Login 5](#_Toc11252557)

[5.2 Asiakkaat 5](#_Toc11252558)

[5.3 Postitoimipaikat 6](#_Toc11252559)

[5.4 Asiakaskategorialuokat 6](#_Toc11252560)

[5.5 Asiakasryhmät 6](#_Toc11252561)

[5.6 Asiakasryhmäluokat 7](#_Toc11252562)

[5.7 Tapahtumat 7](#_Toc11252563)

[5.8 Tapahtumalajit 8](#_Toc11252564)

[5.9 Sähköposti (jatkokehitykseen) 8](#_Toc11252565)

[5.10 Vakioaiheet (liittyy sähköpostiin, jatkokehitykseen) 8](#_Toc11252566)

[6 Luokkakaaviot 9](#_Toc11252567)

[6.1 Luokat 10](#_Toc11252568)

[6.2 Metodit 10](#_Toc11252569)

[7 CRUD-taulukko 11](#_Toc11252570)

[8 Dokumentin versionhallinta 12](#_Toc11252571)

# Järjestelmän tekninen arkkitehtuurikuvaus

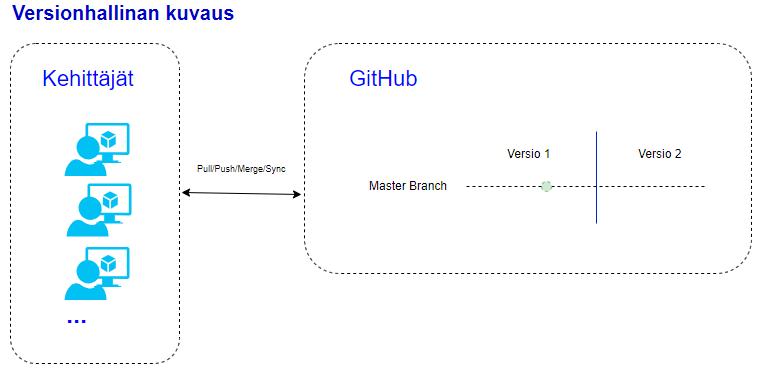
**Yleistä**

Ohjelmistokehityksessä työvälineenä käytetään Microsoft Visual Studiota. Ohjelmointikielenä käytetään C#-kieltä ja ASP.NET:n tarjoamaa MVC-ohjelmistokehystä. Sovelluksen tietokantana käytetään Azuren SQL-Server(Paas) palvelua. Sovelluskehitys tehdään lokaalisti kehittäjien koneilla ja valmis sovellus julkaistaan Azuren App services palveluun.

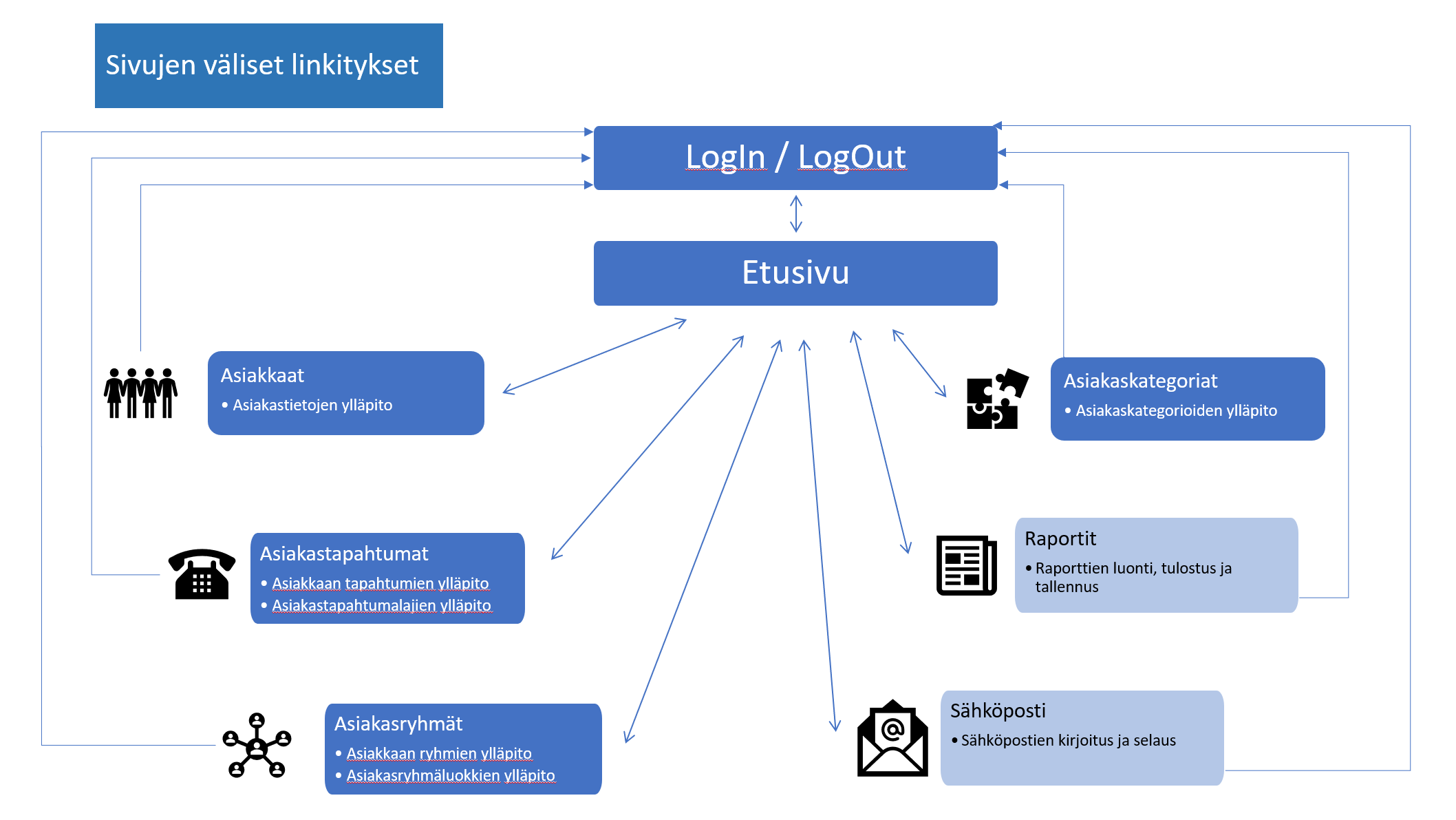
**Versionhallinta**

Versionhallinta toteutetaan Git versionhallintaohjelmistolla, josta koodi siirretään GitHub versionhallintapalveluun. Tätä varten käyttäjille on asennettu Gitin lisäosa Visual Studioon ja luotu tili Github palveluun.

Käyttäjät tekevät kehitystyötä paikallisessa GIT repositoryssä josta se siirretään GitHub versionhallintapalveluun.



# Käyttöliittymä-/sivustokaaviot

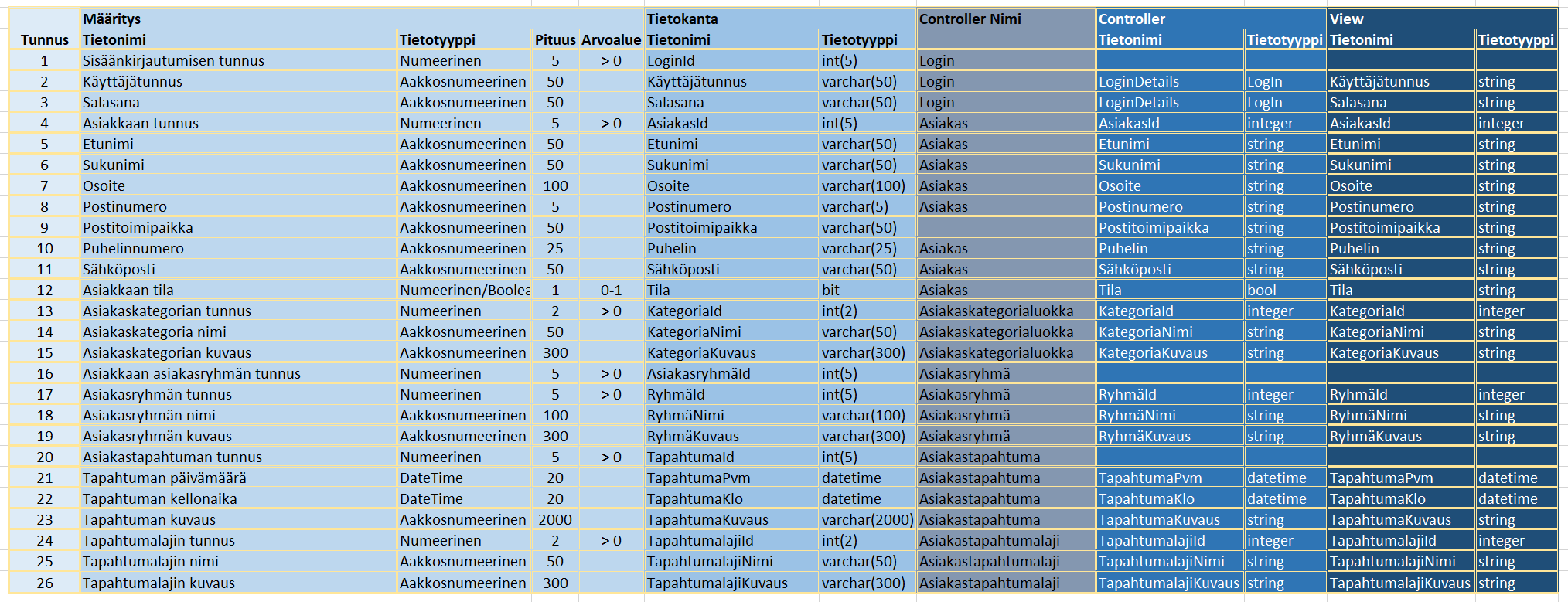


Jatkokehitykseen

Jatkokehitykseen

# Tietohakemisto

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



# Tietokannan ER-malli

Tietokannan ER-malli esitetään tässä kuvana. Sen ylläpito tapahtuu tarvittaessa erillisessä Excel-taulukossa (CRM tietokanta ja tietohakemisto.xlsx), joka on tämän kuvaston liitteenä.



Jatkokehitykseen

# Tietokannan taulujen luontilauseet

## Login

CREATE TABLE [scrum].[Login](

[LoginId] [int] IDENTITY(10000,1) NOT NULL,

[Käyttäjätunnus] [nvarchar](50) NOT NULL,

[Salasana] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Login] PRIMARY KEY CLUSTERED

(

[LoginId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Asiakkaat

CREATE TABLE [scrum].[Asiakkaat](

[AsiakasId] [int] IDENTITY(20000,1) NOT NULL,

[Etunimi] [nvarchar](50) NULL,

[Sukunimi] [nvarchar](50) NULL,

[Osoite] [nvarchar](100) NULL,

[Postinumero] [nvarchar](5) NULL,

[Puhelin] [nvarchar](25) NULL,

[Sähköposti] [nvarchar](50) NULL,

[KategoriaId] [int] NOT NULL,

[Tila] [bit] NOT NULL,

CONSTRAINT [PK\_Asiakkaat] PRIMARY KEY CLUSTERED

(

[AsiakasId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [scrum].[Asiakkaat] WITH NOCHECK ADD CONSTRAINT [FK\_Asiakkaat\_Asiakaskategorialuokat] FOREIGN KEY([KategoriaId])

REFERENCES [scrum].[Asiakaskategorialuokat] ([KategoriaId])

GO

ALTER TABLE [scrum].[Asiakkaat] CHECK CONSTRAINT [FK\_Asiakkaat\_Asiakaskategorialuokat]

GO

ALTER TABLE [scrum].[Asiakkaat] WITH CHECK ADD CONSTRAINT [FK\_Asiakkaat\_Postitoimipaikat] FOREIGN KEY([Postinumero])

REFERENCES [scrum].[Postitoimipaikat] ([Postinumero])

GO

ALTER TABLE [scrum].[Asiakkaat] CHECK CONSTRAINT [FK\_Asiakkaat\_Postitoimipaikat]

GO

## Postitoimipaikat

CREATE TABLE [scrum].[Postitoimipaikat](

[Postinumero] [nvarchar](5) NOT NULL,

[Postitoimipaikka] [nvarchar](50) NULL,

CONSTRAINT [PK\_Postitoimipaikat] PRIMARY KEY CLUSTERED

(

[Postinumero] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Asiakaskategorialuokat

CREATE TABLE [scrum].[Asiakaskategorialuokat](

[KategoriaId] [int] IDENTITY(10,1) NOT NULL,

[KategoriaNimi] [nvarchar](50) NULL,

[KategoriaKuvaus] [nvarchar](300) NULL,

CONSTRAINT [PK\_Asiakaskategorialuokat] PRIMARY KEY CLUSTERED

(

[KategoriaId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Asiakasryhmät

CREATE TABLE [scrum].[Asiakasryhmät](

[AsiakasryhmäId] [int] IDENTITY(97000,1) NOT NULL,

[AsiakasId] [int] NOT NULL,

[RyhmäId] [int] NOT NULL,

CONSTRAINT [PK\_Asiakasryhmät] PRIMARY KEY CLUSTERED

(

[AsiakasryhmäId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [scrum].[Asiakasryhmät] WITH CHECK ADD CONSTRAINT [FK\_Asiakasryhmät\_Asiakasryhmäluokat] FOREIGN KEY([RyhmäId])

REFERENCES [scrum].[Asiakasryhmäluokat] ([RyhmäId])

GO

ALTER TABLE [scrum].[Asiakasryhmät] CHECK CONSTRAINT [FK\_Asiakasryhmät\_Asiakasryhmäluokat]

GO

ALTER TABLE [scrum].[Asiakasryhmät] WITH CHECK ADD CONSTRAINT [FK\_Asiakasryhmät\_Asiakkaat] FOREIGN KEY([AsiakasId])

REFERENCES [scrum].[Asiakkaat] ([AsiakasId])

GO

ALTER TABLE [scrum].[Asiakasryhmät] CHECK CONSTRAINT [FK\_Asiakasryhmät\_Asiakkaat]

GO

## Asiakasryhmäluokat

CREATE TABLE [scrum].[Asiakasryhmäluokat](

[RyhmäId] [int] IDENTITY(40000,1) NOT NULL,

[RyhmäNimi] [nvarchar](100) NULL,

[RyhmäKuvaus] [nvarchar](300) NULL,

CONSTRAINT [PK\_Asiakasryhmäluokat] PRIMARY KEY CLUSTERED

(

[RyhmäId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Tapahtumat

CREATE TABLE [scrum].[Tapahtumat](

[TapahtumaId] [int] IDENTITY(50000,1) NOT NULL,

[AsiakasId] [int] NULL,

[TapahtumalajiId] [int] NULL,

[TapahtumaPvm] [datetime] NULL,

[TapahtumaKlo] [datetime] NULL,

[TapahtumaKuvaus] [nvarchar](2000) NULL,

CONSTRAINT [PK\_Tapahtumat] PRIMARY KEY CLUSTERED

(

[TapahtumaId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [scrum].[Tapahtumat] WITH CHECK ADD CONSTRAINT [FK\_Tapahtumat\_Asiakkaat] FOREIGN KEY([AsiakasId])

REFERENCES [scrum].[Asiakkaat] ([AsiakasId])

GO

ALTER TABLE [scrum].[Tapahtumat] CHECK CONSTRAINT [FK\_Tapahtumat\_Asiakkaat]

GO

ALTER TABLE [scrum].[Tapahtumat] WITH CHECK ADD CONSTRAINT [FK\_Tapahtumat\_Tapahtumalajit] FOREIGN KEY([TapahtumalajiId])

REFERENCES [scrum].[Tapahtumalajit] ([TapahtumalajiId])

GO

ALTER TABLE [scrum].[Tapahtumat] CHECK CONSTRAINT [FK\_Tapahtumat\_Tapahtumalajit]

GO

## Tapahtumalajit

CREATE TABLE [scrum].[Tapahtumalajit](

[TapahtumalajiId] [int] IDENTITY(50,1) NOT NULL,

[TapahtumalajiNimi] [nvarchar](50) NULL,

[TapahtumalajiKuvaus] [nvarchar](300) NULL,

CONSTRAINT [PK\_Tapahtumalajit] PRIMARY KEY CLUSTERED

(

[TapahtumalajiId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Sähköposti (jatkokehitykseen)

CREATE TABLE [scrum].[Sähköposti](

[SähköpostiId] [int] IDENTITY(70000,1) NOT NULL,

[Vastaanottaja] [nvarchar](50) NULL,

[LähetysPvm] [datetime] NULL,

[LähetysKlo] [datetime] NULL,

[AiheId] [int] NULL,

[Otsikko] [nvarchar](100) NULL,

[Viesti] [nvarchar](2000) NULL,

CONSTRAINT [PK\_Sähköposti] PRIMARY KEY CLUSTERED

(

[SähköpostiId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [scrum].[Sähköposti] WITH CHECK ADD CONSTRAINT [FK\_Sähköposti\_Vakioaiheet] FOREIGN KEY([AiheId])

REFERENCES [scrum].[Vakioaiheet] ([AiheId])

GO

ALTER TABLE [scrum].[Sähköposti] CHECK CONSTRAINT [FK\_Sähköposti\_Vakioaiheet]

GO

## Vakioaiheet (liittyy sähköpostiin, jatkokehitykseen)

CREATE TABLE [scrum].[Vakioaiheet](

[AiheId] [int] IDENTITY(95000,1) NOT NULL,

[AiheenNimi] [nvarchar](100) NOT NULL,

CONSTRAINT [PK\_Vakioaiheet] PRIMARY KEY CLUSTERED

(

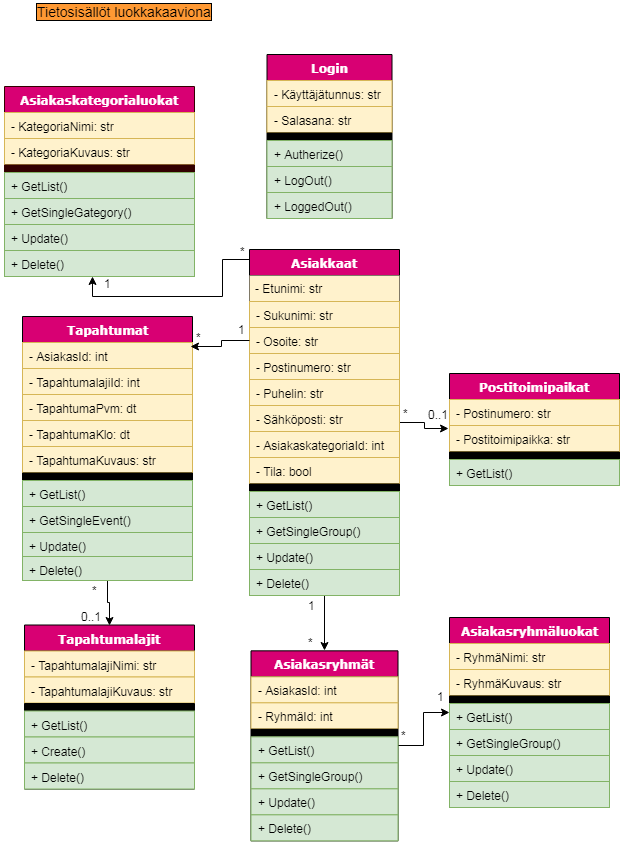
[AiheId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

# Luokkakaaviot



Sovelluksen toiminnot muistuttavat toisiaan eri tietokannan luokissa.

## Luokat

**Login**

* Sisään- ja uloskirjautuminen

**Asiakkaat**

* Asiakastiedot

**Postitoimipaikat**

* Taulussa on listattuna kaikki Suomen postitoimipaikat (v.2015)
* Postitoimipaikat ovat postinumerojärjestyksessä
* Postinumero toimii taulun pääavaimena

**Asiakaskategorialuokat**

* Yksi asiakas kuuluu yhteen ja vain yhteen asiakaskategoriaan
* Asiakaskategoria kuvaa asiakkuuden laatua

**Asiakasryhmäluokat**

* Yksi asiakas voi kuulua useaan asiakasryhmään
* Asiakasryhmiä voidaan käyttää lajittelemaan asiakkaita esim. viiteryhmän mukaan

**Asiakasryhmät**

* Ns. välitaulu, johon kerätään tieto ryhmistä, joihin asiakas on liitetty

**Tapahtumat**

* Asiakkaan kanssa sovittujen toimenpiteiden lisäys, muokkaus ja poisto, esim. päivämäärän tai kellonajan muutos, kuvauksen lisäys/muutos

**Tapahtumalajit**

* Tapahtumalajien ylläpito esim. soitto, tapaaminen, sähköposti

## 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 |
| 0.1 | 1.4.2019 | Dokumentin luonti ja Sprint 1:sen palautukseen valmistelu | Tiimi |
| 0.2 | 14.5.2019 | Tietokannan ER-mallin ja tietohakemiston päivitys DailyScrumissa tehtyjen muutospäätösten perusteella | Tiimi /SKe |
| 1.0 | 10.6.2019 | Päivitetty dokumentti vastaamaan sprint 3:sen lopputilannetta | Tiimi |