



TeeMs

Team Management Software

Janne Kuukkanen

TeeMs

12 2016

Ohjelmistotekniikan ala

Jyväskylän ammattikorkeakoulu

JAMK University of Applied Sciences

Sisältö

1	Asennus	2
2	Tietoa ohjelmasta	2
3	Kuvaruutukaappaukset tärkeimmistä käyttöliittymistä	3
3.1	Login	3
3.2	Register	4
3.3	Home	4
3.4	Profile	5
3.5	Create Project	5
3.6	Project	6
3.7	Create Assignment	6
3.8	Group	7
3.9	View User	7
4	Ohjelman tarvitsemat tiedostot/tietokannat	8
4.1	Tietokannan luonti	8
4.1.1	Skripti	8
4.1.2	Aloitustiedot	9
4.1.3	Kuva tietokannasta	9
5	Tiedossa olevat bugit	10
6	Oppiminen	10
7	Aikataulu	11
	Liitteet	13



1 Asennus

Asennuksessa on huomioitava seuraavat asiat:

Kehityksessä on käytetty seuraavia teknologioita: EntityFramework versio 6.1.3, MySql provider versio 6.9.9.

Mikäli tietokanta on jokin muu kuin kehityksessä käytetty kanta, on muistettava muuttaa TeeMsEntities niminen connectionstring web.config tiedostossa, tämä tapahtuu automaattisesti jos luot uuden entity framework DbContext tiedoston ja asetat sitä kautta uuden yhteyden valitsemaasi tietokantaan, ongelmien välttämiseksi kannattaa poistaa vanha DbContext tiedosto ja nimetä uusi samalla nimellä TeeMs.edmx, Uuden tietokannan täytyy olla MySQL tietokanta ja sen luontiskripti sekä tarvittavat alkudatat on annettu tässä dokumentissa.

2 Tietoa ohjelmasta

toteutetut toiminnalliset vaatimukset:

- Käyttäjän tekeminen (salasanan kryptaaminen)
- Sisäänkirjautuminen
- Aloitus-sivun toiminnallisuus (omien ryhmien ja projektien näyttäminen)
- Uusien projektien luominen
- Uusien ryhmien luominen
- Rymä sivun toiminnallisuus (tietojen näyttäminen ja muokkaaminen tietokannassa, käyttäjien lisääminen ja poistaminen ryhmästä)
- Projekti sivun toiminnallisuus (tietojen näyttäminen ja muokkaaminen tietokannassa, ryhmien lisääminen projektiin)
- Profiilisivun toiminnallisuus (tietojen näyttäminen ja muokkaaminen tietokannassa)
- Muiden käyttäjien profiilin näkeminen

Toteuttamatta jääneet toiminnalliset vaatimukset:

- Projektien sisäisten tehtävien luonti (Assignments)
- Projektien kommentointi
- Chat toiminnallisuus

Ei-toiminnalliset vaatimukset

- Entity frameworkin käyttö
- MySQL tietokanta



Ohjelman dokumentointi löytyy github repositoryn wiki sivulta:

<https://github.com/JanKuukkanen/TeeMS/wiki>.

3 Kuvaruutukaappaukset tärkeimmistä käyttöliittymistä

3.1 Login

A screenshot of the TeeMS login page. The page has a light blue header with the "TeeMS" logo in black and red. Below the header, there is a login form with fields for "Käyttäjänimi:" and "Salasana:". To the right of the "Salasana:" field is a checkbox labeled "Muista tiedot seuraavan kerran." and a "Kirjaudu sisään" button. On the left side of the form, there are links for "Home", "Profile", "Projects", "Groups", "Chat", and "Sign out". At the bottom of the form, there is a link "Don't have an account? Register a new account" and a footer note "Created By: Janne Kuukkanen."

Kuvio 1 Login sivun näkymä

3.2 Register

TeeMs

[Home](#)
[Profile](#)
[Projects](#)
[Groups](#)
[Chat](#)
[Sign out](#)

Sign up for your new account

User Name:
Password:
Confirm Password:
E-mail:
First Name:
Last Name:

Already Registered? [Sign in](#)

Created By: Janne Kuukkanen.

Kuvio 2 Register sivun näkymä

3.3 Home

TeeMs

[Home](#)
[Profile](#)
[Projects](#)
[Groups](#)
[Chat](#)
[Sign out](#)

Your Projects

[Minun projekti](#)

Your Groups

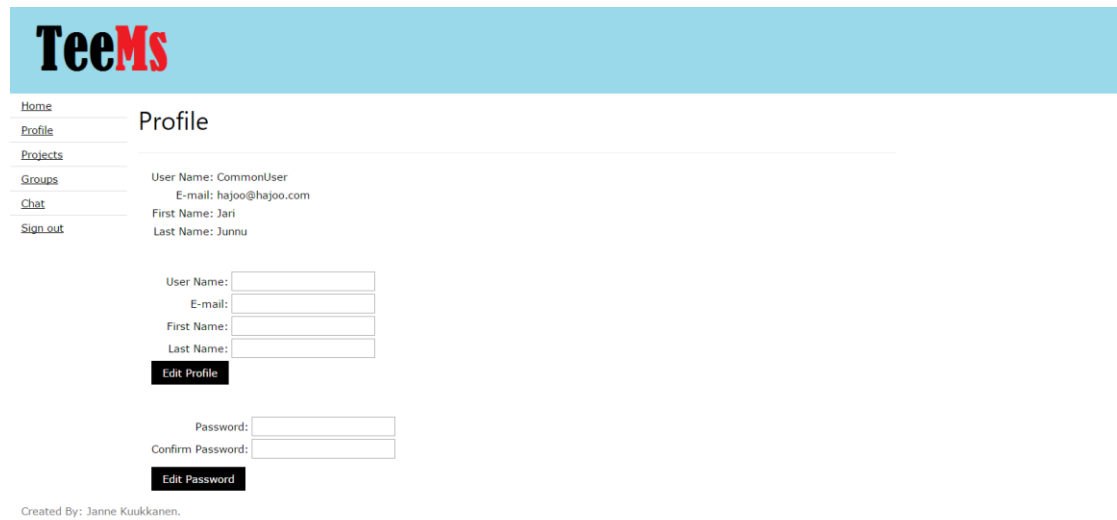
[Text Editors](#)

1

Created By: Janne Kuukkanen.

Kuvio 3 Home sivun näkymä

3.4 Profile



TeeMs

Home
Profile
Projects
Groups
Chat
Sign out

Profile

User Name: CommonUser
E-mail: hajoo@hajoo.com
First Name: Jari
Last Name: Junnu

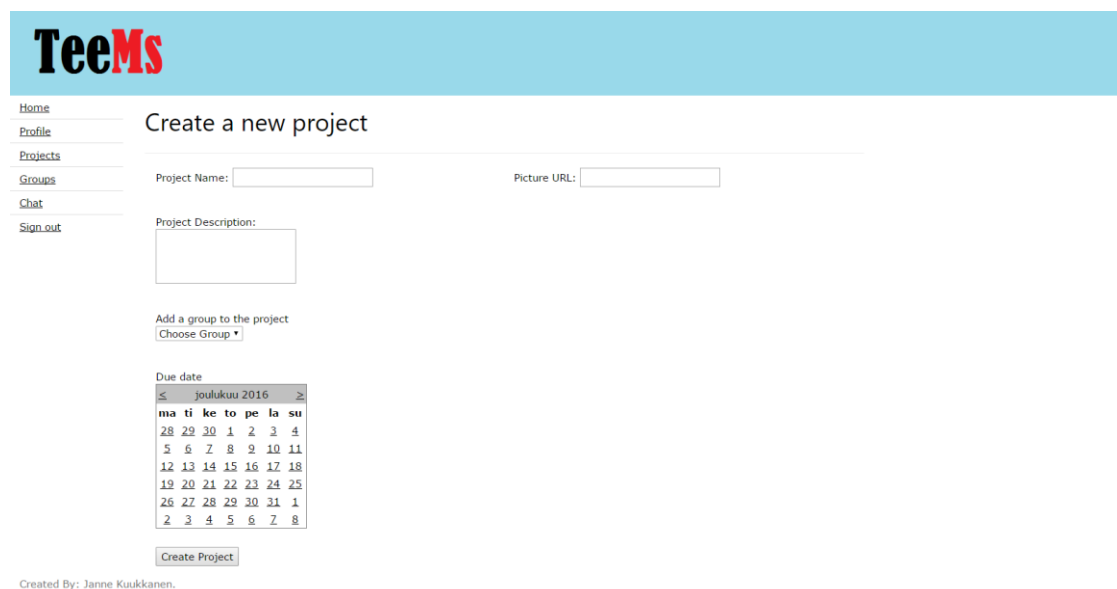
User Name:
E-mail:
First Name:
Last Name:
Edit Profile

Password:
Confirm Password:
Edit Password

Created By: Janne Kuukkanen.

Kuvio 4 Profile sivun näkymä. Tarkoitettu näyttämään ja muokkaamaan käyttäjän omia tietoja.

3.5 Create Project



TeeMs

Home
Profile
Projects
Groups
Chat
Sign out

Create a new project

Project Name: Picture URL:

Project Description:

Add a group to the project
Choose Group ▼

Due date

≤	joulukuu 2016							≥
ma	ti	ke	to	pe	la	su		
28	29	30	1	2	3	4		
5	6	7	8	9	10	11		
12	13	14	15	16	17	18		
19	20	21	22	23	24	25		
26	27	28	29	30	31	1		
2	3	4	5	6	7	8		

Create Project

Created By: Janne Kuukkanen.


Kuvio 5 Create project sivun näkymä

3.6 Project

TeeMs

[Home](#)
[Profile](#)
[Projects](#)
[Groups](#)
[Chat](#)
[Sign out](#)

Minun projekti


Change picture

Project description

Kaikenlaista jännää tulee tapahtumaan tässä projektissa

Edit description

Workflow

Groups working on the project

Choose Group ▾

Close search

Show info

Search

Search

Assignments

Choose Assignment ▾

Create new

Show info

Created By: Janne Kuukkanen.

Kuvio 6 Project sivun näkymä

3.7 Create Assignment

TeeMs

[Home](#)
[Profile](#)
[Projects](#)
[Groups](#)
[Chat](#)
[Sign out](#)

Create a new assignment

Assignment Name:

Assignment Description:

Add a user to the assignment
Choose User ▾

Due date

≤	joulukuu 2016							≥
ma	ti	ke	to	pe	la	su		
28	29	30	1	2	3	4		
5	6	7	8	9	10	11		
12	13	14	15	16	17	18		
19	20	21	22	23	24	25		
26	27	28	29	30	31	1		
2	3	4	5	6	7	8		

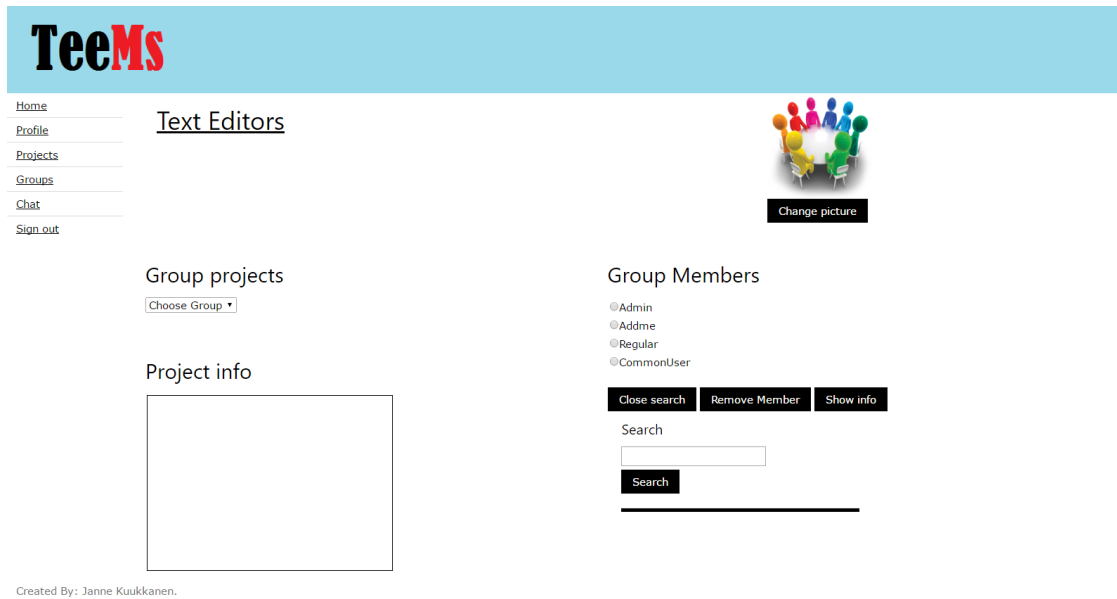
Create Project

1

Created By: Janne Kuukkanen.

Kuvio 7 Create Assignment sivun näkymä

3.8 Group



Kuvio 8 Group sivun näkymä

3.9 View User



Kuvio 3 View User sivun näkymä. tarkoitettu näyttämään muiden ryhmänjäsenten profiilitiedot.

4 Ohjelman tarvitsemat tiedostot/tietokannat

Ohjelman tarvitsemat paketit:

- EntityFramework.6.1.3
- MySql.Data.6.9.9
- MySql.Data.Entity.6.9.9
- Newtonsoft.Json.9.0.1

Bin-tiedoston sisältö:

- EntityFramework.dll
- EntityFramework.SqlServer.dll
- MySql.Data.dll
- MySql.Data.Entity.EF6.dll
- Newtonsoft.Json.dll

Image-kansion kuvat:

- MP_Logo.png
- MP_Logo_alt.png
- no_image.png

4.1 Tietokannan luonti

4.1.1 Skripti

Katso liite: TeeMs_database_creation_script.txt

4.1.2 Aloitus data

Nämä tiedot on syötettävä TeeMs:n ennen käyttöä

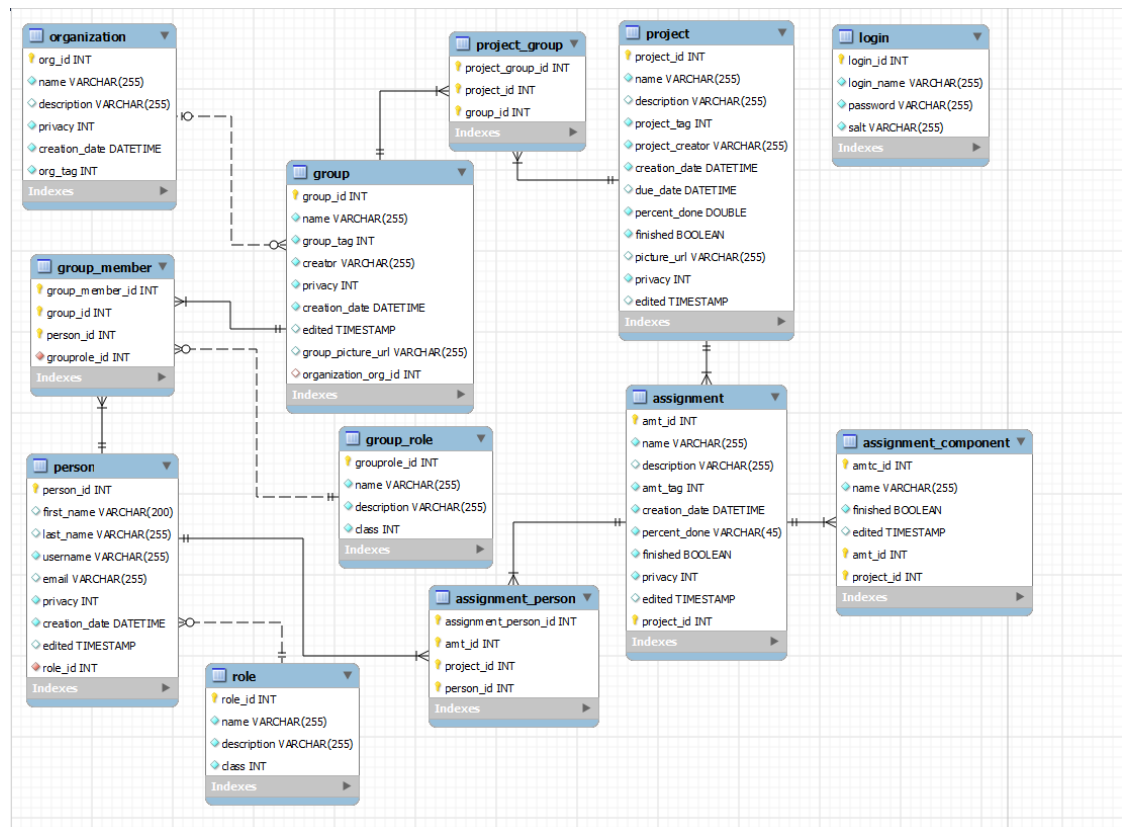
User roles:

```
INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES
('Administrator', 'Highest Possible rank, only one should exist at
any given time', '1');
INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES
('Moderator', 'Second highest rank, can delete basic users', '2');
INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES
('User', 'Basic user rank, given to anyone upon completing registra-
tion', '3');
```

Group_Member roles:

```
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`)
VALUES ('Group Member', 'Basic member of the group', '4');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`)
VALUES ('Group Moderator', 'Promoted member of the group', '3');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`)
VALUES ('Group Administrator', 'Group leader', '2');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`)
VALUES ('Group Creator', 'Original creator of the group', '1');
```

4.1.3 Kuva tietokannasta



5 Tiedossa olevat bugit

Create Assignment sivulla käyttäjien listaus dropdownlistissa ei toimi, lista näyttää vain yhden käyttäjän.

Jatkossa kehitystä tullaan jatkamaan lisäämällä kaikki kohdassa 2 listatut toteuttamattomat ominaisuudet, sen lisäksi ohjelmaan tulisi implementoida ylläpitoon liittyvä käyttörajapinta (eli ns. Admin näkymä), käyttäjien roolia ja yksityisyyttä hyödyntäviä ominaisuuksia voidaan myös tulla lisäämään.

Chat ominaisuus ja projektien kommentoinnin voisi toteuttaa XML-tiedostoilla.

6 Oppiminen

Työtä tehdessä oppi erityisesti Entity frameworkin käytön sekä eri ASP:NET kontrollien ominaisuuksien hyödyntämisen sivua tehdessä. Myös salasanojen kryptaus ja suolaus tuli toteutettua ensimmäisen kerran tätä työtä tehdessä ja näiden toteutus antoi vastauksen muutamiin kysymyksiin jotka olivat häirinneet käyttäjän profiilitietojen säilytyksessä jo jonkin aikaa.

Suurimmat haasteet olivat Gridviewn upottaminen haku-ominaisuuden osaksi sillä sen eri kenttien muokkaaminen osoittautui todella työlääksi, myös arvon saaminen gridviewn solusta oli erikoisen komennon takana, tosin tämä johtui gridviewn omalaatuisesta toteutuksesta.

Muita haasteita olivat entity frameworkin konteksti tiedoston kanssa taistelu jos teki jonkinlaisia muutoksia tietokantaan.

Enemmän tutkittavaa olisi erilaisissa toteutuksissa haku toiminnon aikaansaamiseksi, ja on entity frameworkissakin vielä tutkimista.

7 Aikataulu

10.10.2016 selite: Ohjelmiston määrittely ja konsepti kaavio
H: 3

12.10.2016 selite: Avattu Trello lauta työskentelyn organisoimista varten, alustavasti käyty läpi tehtäviä
H: 1.5

27.10.2016 selite: Mockuppeja tuotteen UI:sta sekä Github wiki:n ja repositoryn luonti
H: 3.25

15.11.2016 selite: Mockuppien ja työn kuvauksen tarkennusta
H: 1

16.11.2016 selite: Mockuppien ja työn kuvauksen tarkennusta, Tietokannan suunnittelu
H: 4

17.11.2016 selite: Mockuppien tarkennusta ja tietokannan suunnittelu
H: 3

18.11.2016 selite: Mockuppien tarkennusta ja tietokannan suunnittelu
H: 1.75

20.11.2016 selite: Tietokanta skeeman teko ja TeeMS projektin luonti
H: 3

21.11.2016 selite: aihe-ehdotuksen tekeminen ja tietokannan luominen palvelimella
H: 2.25

23.11.2016 selite: Home-sivun tekemistä
H: 4

25.11.2016 selite: Verkkosivuston perusrungon luonti
H: 1

28.11.2016 selite: Sisäänkirjautumisen implementointi
H: 7.75

29.11.2016 selite: Sisään- ja uloskirjautumisen implementointi, Uuden ryhmän luonnin ohjelmointi



H: 7.5

30.11.2016 selite: Ryhmän luonnin ohjelmointi sekä profile sivun luonti

H: 7.5

1.12.2016 selite: Profile ja ryhmä sivun ohjelmointi

H: 5

2.12.2016 selite: Projektin luonti sivun ohjelmointi sekä ryhmäluonnin parantelua

H: 5.25

3.12.2016 selite: Projektin luonnin ohjelmointi ja Home sivun toiminnallisuuden ohjelmointi

H: 5

4.12.2016 selite: Projekti sivun ohjelmointi ja yeisiä parannuksia

H: 5.5

5.12.2016 selite: Ryhmäjäsänen lisäämisen toiminnallisuuden ohjelmointi

H: 6.25

6.12.2016 selite: Ryhmä ja projekti sivun toiminnallisuuden parantamista

H: 7

7.12.2016 selite: Projektitehtävän luonti sivun tekeminen

H: 4

Kokonaistuntimäärä: 81 tuntia

Arvosana ehdotus: 3



Liitteet

Liite 1. TeeMs_database_creation_script.txt

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `H3387` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `H3387` ;

-- -----
-- Table `H3387`.`role`
-- -----
CREATE TABLE IF NOT EXISTS `H3387`.`role` (
  `role_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(255) NOT NULL ,
  `description` VARCHAR(255) NOT NULL ,
  `class` INT NOT NULL ,
  PRIMARY KEY (`role_id`) )
ENGINE = InnoDB;

-- -----
-- Table `H3387`.`person`
-- -----
CREATE TABLE IF NOT EXISTS `H3387`.`person` (
  `person_id` INT NOT NULL AUTO_INCREMENT ,
  `first_name` VARCHAR(200) NULL ,
  `last_name` VARCHAR(255) NULL ,
  `username` VARCHAR(255) NOT NULL ,
  `email` VARCHAR(255) NULL ,
  `privacy` INT NOT NULL ,
  `creation_date` DATETIME NOT NULL ,
  `edited` TIMESTAMP NULL ,
  `role_id` INT NOT NULL ,
  PRIMARY KEY (`person_id`) ,
  INDEX `fk_person_role1_idx` (`role_id` ASC) ,
  UNIQUE INDEX `username_UNIQUE` (`username` ASC) ,
  CONSTRAINT `fk_person_role1`
    FOREIGN KEY (`role_id`)
      REFERENCES `H3387`.`role` (`role_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `H3387`.`project`
```



```
-----  
CREATE TABLE IF NOT EXISTS `H3387`.`project` (  
  `project_id` INT NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(255) NOT NULL ,  
  `description` VARCHAR(255) NULL ,  
  `project_tag` INT NOT NULL ,  
  `project_creator` VARCHAR(255) NOT NULL ,  
  `creation_date` DATETIME NOT NULL ,  
  `due_date` DATETIME NULL ,  
  `percent_done` DOUBLE NOT NULL ,  
  `finished` TINYINT(1) NOT NULL ,  
  `picture_url` VARCHAR(255) NULL ,  
  `privacy` INT NOT NULL ,  
  `edited` TIMESTAMP NULL ,  
  PRIMARY KEY (`project_id`) ,  
  UNIQUE INDEX `project_tag_UNIQUE` (`project_tag` ASC) )  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `H3387`.`organization`  
-----
```

```
CREATE TABLE IF NOT EXISTS `H3387`.`organization` (  
  `org_id` INT NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(255) NOT NULL ,  
  `description` VARCHAR(255) NULL ,  
  `privacy` INT NOT NULL ,  
  `creation_date` DATETIME NOT NULL ,  
  `org_tag` INT NOT NULL ,  
  PRIMARY KEY (`org_id`) ,  
  UNIQUE INDEX `org_tag_UNIQUE` (`org_tag` ASC) )  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `H3387`.`group`  
-----
```

```
CREATE TABLE IF NOT EXISTS `H3387`.`group` (  
  `group_id` INT NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(255) NOT NULL ,  
  `group_tag` INT NOT NULL ,  
  `creator` VARCHAR(255) NOT NULL ,  
  `privacy` INT NOT NULL ,  
  `creation_date` DATETIME NOT NULL ,  
  `edited` TIMESTAMP NULL ,  
  `group_picture_url` VARCHAR(255) NULL ,  
  `organization_org_id` INT NULL ,  
  PRIMARY KEY (`group_id`) ,  
  INDEX `fk_group_organization_idx` (`organization_org_id` ASC) ,  
  UNIQUE INDEX `group_tag_UNIQUE` (`group_tag` ASC) ,  
  CONSTRAINT `fk_group_organization`  
    FOREIGN KEY (`organization_org_id` )  
    REFERENCES `H3387`.`organization` (`org_id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```



```
-----
-- Table `H3387`.`assignment`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`assignment` (
  `amt_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(255) NOT NULL ,
  `description` VARCHAR(255) NULL ,
  `amt_tag` INT NOT NULL ,
  `creation_date` DATETIME NOT NULL ,
  `percent_done` VARCHAR(45) NOT NULL ,
  `finished` TINYINT(1) NOT NULL ,
  `privacy` INT NOT NULL ,
  `edited` TIMESTAMP NULL ,
  `project_id` INT NOT NULL ,
  PRIMARY KEY (`amt_id`, `project_id`) ,
  INDEX `fk_assignment_project1_idx` (`project_id` ASC) ,
  UNIQUE INDEX `amt_tag_UNIQUE` (`amt_tag` ASC) ,
  CONSTRAINT `fk_assignment_project1`
    FOREIGN KEY (`project_id`)
      REFERENCES `H3387`.`project` (`project_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `H3387`.`assignment_component`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`assignment_component` (
  `amtc_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(255) NOT NULL ,
  `finished` TINYINT(1) NOT NULL ,
  `edited` TIMESTAMP NULL ,
  `amt_id` INT NOT NULL ,
  `project_id` INT NOT NULL ,
  PRIMARY KEY (`amtc_id`, `amt_id`, `project_id`) ,
  INDEX `fk_assignment_component_assignment1_idx` (`amt_id` ASC,
`project_id` ASC) ,
  CONSTRAINT `fk_assignment_component_assignment1`
    FOREIGN KEY (`amt_id`, `project_id`)
      REFERENCES `H3387`.`assignment` (`amt_id`, `project_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `H3387`.`login`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`login` (
  `login_id` INT NOT NULL AUTO_INCREMENT ,
  `login_name` VARCHAR(255) NOT NULL ,
  `password` VARCHAR(255) NOT NULL ,
  `salt` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`login_id`) ,
  UNIQUE INDEX `username_UNIQUE` (`login_name` ASC) )
ENGINE = InnoDB;
```




```
-----
-- Table `H3387`.`group_role`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`group_role` (
  `grouprole_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(255) NOT NULL ,
  `description` VARCHAR(255) NOT NULL ,
  `class` INT NOT NULL ,
  PRIMARY KEY (`grouprole_id`) )
ENGINE = InnoDB;

-----
-- Table `H3387`.`group_member`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`group_member` (
  `group_member_id` INT NOT NULL AUTO_INCREMENT ,
  `group_id` INT NOT NULL ,
  `person_id` INT NOT NULL ,
  `grouprole_id` INT NOT NULL ,
  PRIMARY KEY (`group_member_id`, `group_id`, `person_id`) ,
  INDEX `fk_group_has_person_person1_idx` (`person_id` ASC) ,
  INDEX `fk_group_has_person_group1_idx` (`group_id` ASC) ,
  INDEX `fk_group_member_group_role1_idx` (`grouprole_id` ASC) ,
  CONSTRAINT `fk_group_has_person_group1`
    FOREIGN KEY (`group_id`)
      REFERENCES `H3387`.`group` (`group_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_group_has_person_person1`
    FOREIGN KEY (`person_id`)
      REFERENCES `H3387`.`person` (`person_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_group_member_group_role1`
    FOREIGN KEY (`grouprole_id`)
      REFERENCES `H3387`.`group_role` (`grouprole_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `H3387`.`project_group`
-----
CREATE TABLE IF NOT EXISTS `H3387`.`project_group` (
  `project_group_id` INT NOT NULL AUTO_INCREMENT ,
  `project_id` INT NOT NULL ,
  `group_id` INT NOT NULL ,
  PRIMARY KEY (`project_group_id`, `project_id`, `group_id`) ,
  INDEX `fk_project_has_group_group1_idx` (`group_id` ASC) ,
  INDEX `fk_project_has_group_project1_idx` (`project_id` ASC) ,
  CONSTRAINT `fk_project_has_group_project1`
    FOREIGN KEY (`project_id`)
      REFERENCES `H3387`.`project` (`project_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```



```
CONSTRAINT `fk_project_has_group_group1`  
  FOREIGN KEY (`group_id` )  
  REFERENCES `H3387`.`group` (`group_id` )  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `H3387`.`assignment_person`  
-----
```

```
CREATE TABLE IF NOT EXISTS `H3387`.`assignment_person` (  
  `assignment_person_id` INT NOT NULL AUTO_INCREMENT ,  
  `amt_id` INT NOT NULL ,  
  `project_id` INT NOT NULL ,  
  `person_id` INT NOT NULL ,  
  PRIMARY KEY (`assignment_person_id`, `amt_id`, `project_id`,  
  `person_id` ) ,  
  INDEX `fk_assignment_has_person_person1_idx` (`person_id` ASC) ,  
  INDEX `fk_assignment_has_person_assignment1_idx` (`amt_id` ASC,  
  `project_id` ASC) ,  
  CONSTRAINT `fk_assignment_has_person_assignment1`  
    FOREIGN KEY (`amt_id` , `project_id` )  
    REFERENCES `H3387`.`assignment` (`amt_id` , `project_id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_assignment_has_person_person1`  
    FOREIGN KEY (`person_id` )  
    REFERENCES `H3387`.`person` (`person_id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
USE `H3387` ;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```