– E4 –

Projet Personnel Encadré

– Sleipnir –

# Situation initiale

La société Kronos, start-up spécialisée dans la location de matériel informatique, gère actuellement ses stocks grâce à un logiciel payant. Celui-ci n’est pas accessible en dehors de la machine sur laquelle il est installé. Kronos souhaite donc faire développer une plate-forme web de gestion d’équipements, accessible depuis n’importe quel ordinateur de la société, via leur Intranet, gratuitement. La société confie ce projet à un alternant recruté récemment, lequel nommera le projet *Sleipnir*.

# 

# Solution proposée

L’alternant décide de se limiter à des langages de programmation simples, sans l’aide d’aucun framework, afin que la plate-forme soit maintenable facilement sans lui. Ainsi, il utilisera le trio HTML/PHP/CSS et un serveur Ubuntu installé sur une machine visible sur le réseau Intranet de l’entreprise. La première chose à faire est donc d’installer puis de paramétrer ce serveur afin de pouvoir développer en s’y connectant.

# Paramétrage d’Ubuntu Server

Une fois installé, certains packages sont nécessaires afin de pouvoir développer sur Ubuntu Server :

* SSH
* apache2
* mysql-server
* php7
* phpmyadmin
* proftpd

Les chemins de fichiers de configuration importants sont :

/etc/apache2/apache2.conf

/etc/php/7.0/apache2/php.ini

Il est important de faire un lien symbolique du dossier phpmyadmin vers html, comme suit :

cd /var/www/html

ln -s /usr/share/phpmyadmin

Cela permet d’accéder à phpmyadmin via ‘*adresse\_ip/phpmyadmin*’.

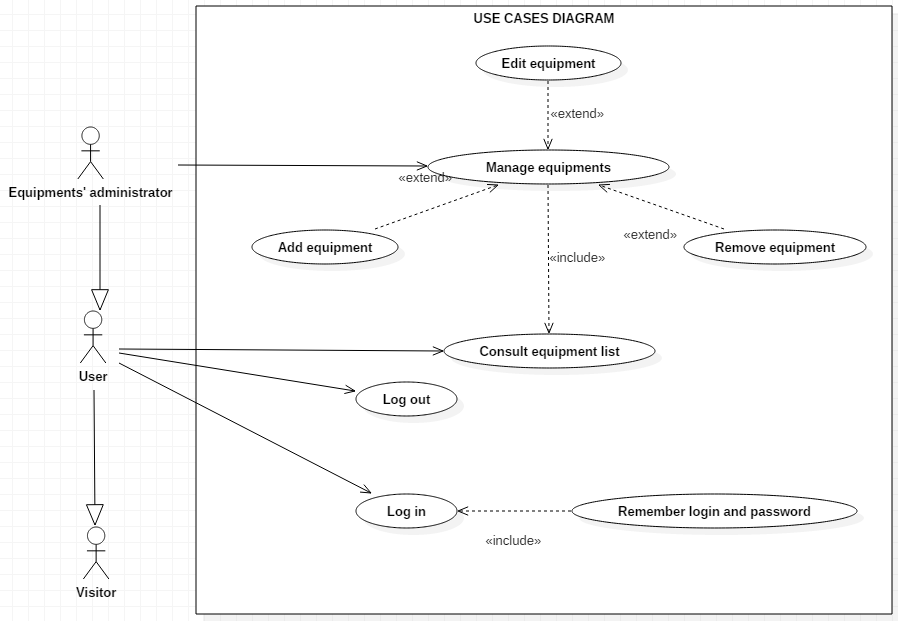
L’installation et la configuration terminées, le site peut commencer à être modélisé, et nous utiliserons pour cela le SysML, modelisé via le logiciel gratuit StarUML.

# Modélisation SysML

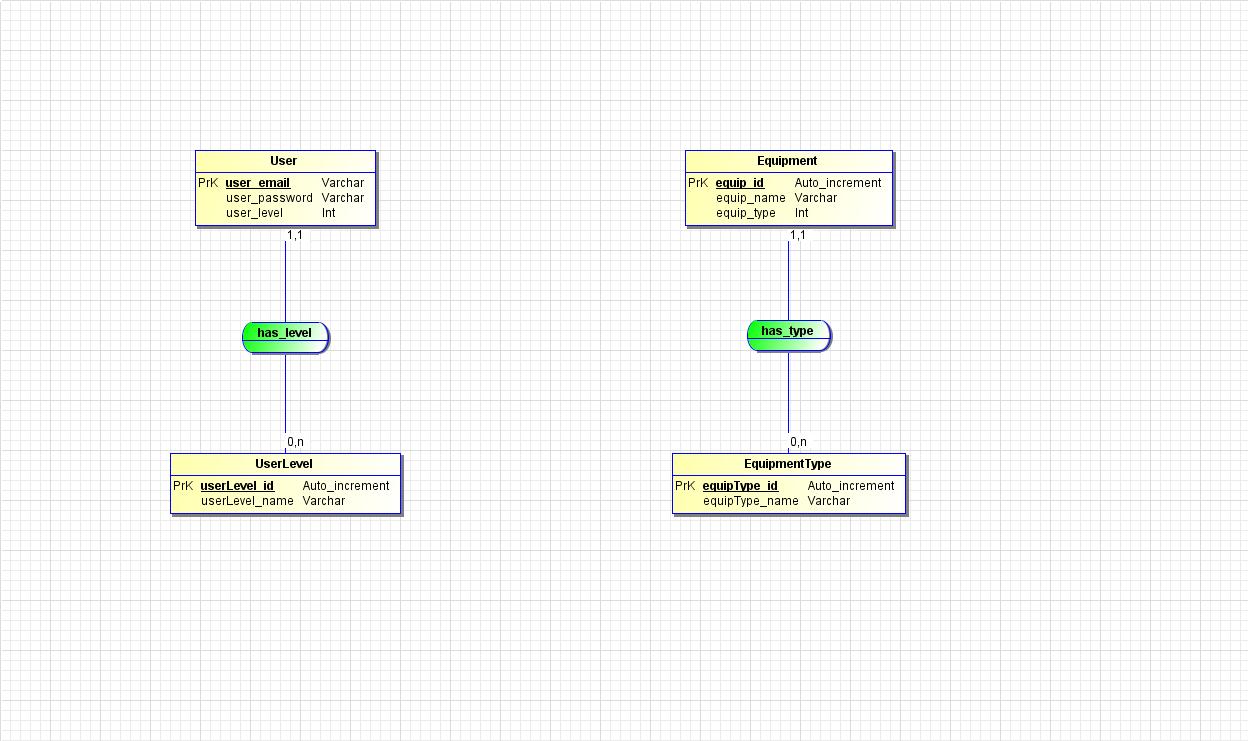
## Diagramme de cas d’utilisations

Il y aura donc trois types d’utilisateurs :

* Le visiteur, ne possédant pas de compte, n’aura aucun droit ni de consultation, ni de modification.
* L’utilisateur, possédant un compte classique, aura le droit de consulter la liste des équipements uniquement.
* L’administrateur, quant à lui, aura le droit de consulter la liste des équipements, mais aussi de modifier ceux-ci. Les trois types de modifications sont l’ajout, l’édition, et la suppression.

Les utilisateurs, aux nombres donc de deux, seront stockés dans la base de données. Ces comptes permettront de gérer les droits accordés aux employés pouvant se connecter au site.

# Base de données



La base de données est modélisée via JMerise. Elle est relativement basique, et ne possède que le minimum possible d’informations nécessaires au bon fonctionnement de la plate-forme Sleipnir.

Les utilisateurs possèderont une clé primaire consistant en une adresse e-mail, qui servira de nom d’utilisateur ; d’un mot de passe de connexion ; et enfin d’un niveau d’utilisateur, possiblement utilisable lorsque le besoin de créer différents niveaux de droits et/ou différents utilisateurs s’en fera sentir.

La deuxième table comprendra la liste des équipements, consistant en un ID, un nom, et un type.

Les deux requêtes de création se feront directement par le biais de phpmyadmin, et seront liées à des fichiers .CSV, contenant la liste des utilisateurs dans l’un, et une liste simple d’équipements dans l’autre. Cela permettra d’avoir une plate-forme fonctionnelle dès la mise en place du site.

Voici ces deux requêtes, ainsi que le contenu des quatre fichiers .csv vers lesquels elles pointent.

#------------------------------------------------------------

# Script MySQL.

#------------------------------------------------------------

CREATE database IF NOT EXISTS `sleipnir\_users`;

USE `sleipnir\_users`;

#------------------------------------------------------------

# Table: User

#------------------------------------------------------------

CREATE TABLE IF NOT EXISTS `User`(

user\_email Varchar (100) NOT NULL,

user\_password Varchar (64) NOT NULL,

FK\_userLevel\_id Int NOT NULL,

PRIMARY KEY (user\_email)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: UserLevel

#------------------------------------------------------------

CREATE TABLE IF NOT EXISTS `UserLevel`(

userLevel\_id int (11) Auto\_increment NOT NULL,

userLevel\_name Varchar (128) NOT NULL,

PRIMARY KEY (userLevel\_id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Foreign Key attribute adding

#------------------------------------------------------------

ALTER TABLE User ADD CONSTRAINT FK\_User\_userLevel\_id FOREIGN KEY (FK\_userLevel\_id) REFERENCES UserLevel(userLevel\_id);

#------------------------------------------------------------

# Initial CSV data import

#------------------------------------------------------------

LOAD DATA LOCAL INFILE '/var/www/html/include/database/resources/csv/UserLevel.csv'

INTO TABLE UserLevel

CHARACTER SET utf8

FIELDS TERMINATED BY ','

IGNORE 1 LINES

(UserLevel.userLevel\_id, UserLevel.userLevel\_name);

LOAD DATA LOCAL INFILE '/var/www/html/include/database/resources/csv/User.csv'

INTO TABLE User

CHARACTER SET utf8

FIELDS TERMINATED BY ','

IGNORE 1 LINES

(User.user\_email, User.user\_password, User.FK\_userLevel\_id);

#------------------------------------------------------------

# Script MySQL.

#------------------------------------------------------------

CREATE database IF NOT EXISTS `sleipnir\_equipments`;

USE `sleipnir\_equipments`;

#------------------------------------------------------------

# Table: Equipment

#------------------------------------------------------------

CREATE TABLE IF NOT EXISTS `Equipment`(

equip\_id int (11) Auto\_increment NOT NULL,

equip\_name Varchar (255) NOT NULL,

FK\_equipType\_id Int NOT NULL,

PRIMARY KEY (equip\_id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: EquipmentType

#------------------------------------------------------------

CREATE TABLE IF NOT EXISTS `EquipmentType`(

equipType\_id int (11) Auto\_increment NOT NULL,

equipType\_name Varchar (255) NOT NULL,

PRIMARY KEY (equipType\_id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Foreign Key attribute adding

#------------------------------------------------------------

ALTER TABLE Equipment ADD CONSTRAINT FK\_Equipment\_equipType\_id FOREIGN KEY (FK\_equipType\_id) REFERENCES EquipmentType(equipType\_id);

#------------------------------------------------------------

# Initial CSV data import

#------------------------------------------------------------

LOAD DATA LOCAL INFILE '/var/www/html/include/database/resources/csv/EquipmentType.csv'

INTO TABLE EquipmentType

CHARACTER SET utf8

FIELDS TERMINATED BY ','

IGNORE 1 LINES

(EquipmentType.equipType\_id, EquipmentType.equipType\_name);

LOAD DATA LOCAL INFILE '/var/www/html/include/database/resources/csv/Equipment.csv'

INTO TABLE Equipment

CHARACTER SET utf8

FIELDS TERMINATED BY ','

IGNORE 1 LINES

(Equipment.equip\_name, Equipment.FK\_equipType\_id);

* User.csv

user\_email,user\_password,FK\_userLevel\_id

sleipnir\_admin@sleipnir.com,e652ceecbe6ea0496fa226fcee823eaa8d1dc63f80334966b0cf0999131c7f7e,1

user1@sleipnir.com,12654c024a4c0926329753cf79d4bb95d617c17684066809cf453b2084a4d5bc,2

Les mots des passe sont encodes en sha256, et sont, une fois décodés :

Sleipnir\_admin@sleipnir.com : GnC3BRbkt5kd

user1@sleipnir.com : userpass1

* UserLevel.csv

userLevel\_id,userLevel\_name

1,admin

2,user

3,visitor

* Equipment.csv

equip\_name,FK\_equipType\_id

asus\_pc\_1,1

asus\_pc\_2,1

asus\_laptop\_1,2

asus\_laptop\_2,2

printer\_1,3

printer\_2,3

* EquipmentType.csv

equipType\_id,equipType\_name

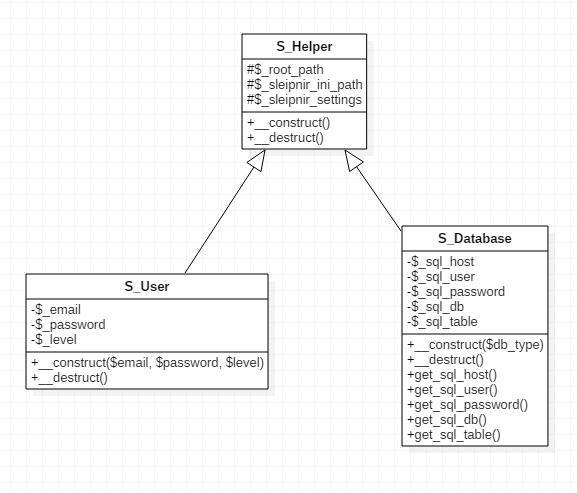
1,computer

2,laptop

3,printer

# Helper

Le fichier helper.php contiendra des classes et fonctions d’aide au codage.



S\_Helper possède les chemins d’accès importants, soient celui de la racine du site, et celui de *‘sleipnir\_settings.ini’,* contenant les informations de configuration et d’authentification à mysql et Sleipnir.

S\_User et S\_Database, quant à elles, héritent de S\_Helper.

S\_Database est construite de façon à aller chercher des informations dans le fichier de configuration suivant le type de base, donné en paramètre lors de sa création. Il est ainsi possible de rapidement et simplement se connecter à la base de données voulue.

S\_User servira lors de la gestion de multiples utilisateurs.

## Fonctions de soutien

Helper.php contient également deux fonctions de soutien, l’une permettant de sécuriser une chaîne de caractères, utilisée lors d’une requête à une table utilisant des données remplies par l’utilisateur, afin d’éviter les injections SQL, et l’autre permettant de séparer les données d’un tableau donné en <option></option>, permettant de créer facilement des boites à choix multiples.