Robin TERSOU, Antoine CHEVAL DARRACQ, Angelo DELIESSCHE

ESGI | 242 RUE DU FAUBOURG SAINT-ANTOINE, PARIS 75012

Documentation Technique et fonctionnelle

My corp guardian

# **Contexte et objectifs**

Le but de ce projet, est de mettre en place une architecture logicielle complète permettant à un client (entreprise ou particulier), de pouvoir gérer entièrement le contrôle d’accès de son bâtiment.

Pour ce faire, nous allons mettre en place deux applications ayant chacune leurs fonctionnalités. Celles-ci seront connectées à une API commune utilisant des technologies récentes (NodeJS).

Parmi ces deux applications, la première sera développée en JAVA et la seconde en langage Web. Nous avons prévu, pour l’application JAVA, une architecture complètement évolutive permettant d’accueillir de nombreux plugins. Ces plugins permettront d’apporter des fonctionnalités supplémentaires au logiciel (détections de caméras, de détecteurs, nouvelles alarmes etc..). Ainsi, le client pourra faire évoluer lui-même le logiciel en développant, par exemple, ses propres plugins.

Pour ce qui est de la deuxième application, elle sera accessible par un navigateur web (à conditions de posséder les identifiants de connexion). Son but est simple : proposer une interface sobre et rapide d’utilisation pour accéder à certaines fonctionnalités du logiciel, à distance. Ainsi, elle permettra de visionner un live des caméras lors d’une détection ou d’un incident par exemple. De plus, une page de statistiques sera disponible via cette application web.

# **Charte graphique et logo**

## Logo



## Polices

|  |  |  |  |
| --- | --- | --- | --- |
| Types de textes | polices | Taille | Couleur |
| Titres (h1, h2) |  |  |  |
| Sous-titres (h3,h4) |  |  |  |
| Liens |  |  |  |
| paragraphes |  |  |  |
| labels |  |  |  |

## Couleurs

* Palette de couleurs principales
* Palette de couleurs secondaires

# **Liste des fonctionnalités**

## Application JAVA

### Authentification

* + Connexion / Déconnexion
  + Mot de passe oublié

### Gestion des groupes d’utilisateur

* + Créer / Modifier / Supprimer un groupe
  + Affecter une tranche horaire

### Gestion des utilisateurs

* + Créer / Modifier / Supprimer un utilisateur
  + Affecter un groupe
  + Affecter / désaffecter un badge
  + Affecter une tranche horaire

### Gestion des tranches horaires

* + Créer / Modifier / Supprimer une tranche horaire

### Gestion du matériel

* + Créer / Modifier / Supprimer
  + Affichage des informations du matériel
  + Assigner une tranche horaire
  + Assigner un groupe d’utilisateur

### Gestion des systèmes de sécurité

* + Armer / Désarmer le système
  + Affichage du matériel installé
  + Affichage de l’historique des événements
  + Affichage des retours des caméras
  + Modélisation et affichage du bâtiment modélisé

### Gestion des plugins

* + Ajouter / Supprimer un plugin
  + Consulter les plugins

## Application WEB

Armer desarmer

Flux camera

Infos des utilisateurs

Compteur

Historique des evenements

# **Descriptif fonctionnel**

## Acteurs et architecture globale

Base de données

INTERFACE

WEB

APPLICATION

JAVA

**Administrateur**

## Application JAVA

### Authentification

**Description :**

Au lancement de l’application, l’administrateur se verra afficher la page d’accueil lui demandant de s’authentifier. L’administrateur aura donc 2 options :

* + - Connexion par identifiant / Mot de passe

Lorsque la connexion est réussie, un Token de connexion est distribué à **l’administrateur**. Ce token permettra de se connecter à l’API et d’accéder aux différentes fonctionnalités de l’application.

* Connexion par identifiants

Lorsqu’un **administrateur** est créé, il se voit attribué un login et un mot de passe. Ceux-ci lui permettent de se connecter à l’application.

**Déroulement :**

**Situation initiale :** Affichage de la page de connexion par identifiants à l’administrateur.

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Remplissage des champs de formulaire affichés par le système et validation du formulaire. | 10 |  |
|  | 20 | Vérification effectuée et validée sur les champs : **#VERA01** |
|  | 30 | Attribution de l’identifiant de connexion |
|  | 40 | Affichage de la page d’accueil de l’application |

Scénario alternatif

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 21 | Vérification échouée : **#VERA01** |
|  | 22 | Retour à la situation initiale.  Affichage des messages d’erreurs **#ERRA01** correspondants. |

**Situation finale (Scénario nominal) ;** Affichage de la page d’accueil de l’application, l’administrateur est authentifié.

Un jeton de connexion est donc attribué à l’administrateur connecté. Ce jeton sera utilisé pour chaque requête sur l’API.

* Connexion par Badge

Si un plugin permettant de se connecter via la passage d’un badge est déjà installé, l’administrateur aura la possibilité de se connecter avec le badge qu’il a pu s’associer grâce à la fonctionnalité présente sur le logiciel.

**Déroulement :**

**Situation initiale :** Affichage de la page de connexion à l’administrateur

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Vérification effectuée et validée sur la présence du plugin de connexion par badge : **#VERP01** |
|  | 20 | Ecoute constante jusqu’au passage d’un badge par l’administrateur |
| Passage du badge par l’administrateur | 30 |  |
|  | 40 | Vérification effectuée et validée sur les identifiants du badge : **#VERA02** |
|  | 50 | Attribution de l’identifiant de connexion |
|  | 60 | Affichage de la page d’accueil |

Scénario alternatif A :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 11 | Vérification échouée sur la présence du plugin de connexion par badge : **#VERP01** |
|  | 12 | Retour à la situation initiale.  Affichage du message d’erreur **#ERRA020** |

Scénario alternatif B :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 41 | Vérification échouée sur les identifiants de connexions : **#VERA02** |
|  | 42 | Retour à la situation initiale.  Affichage du message d’erreur **#ERRA021** |

**Situation finale (Scénario nominal) :** Affichage de la page d’accueil de l’application, l’administrateur est authentifié.

Un jeton de connexion est donc attribué à l’administrateur connecté. Ce jeton sera utilisé pour chaque requête sur l’API.

* Mot de passe oublié

Sur la page de connexion, il apparaît une option à l’administrateur indiquant : « Mot de passe oublié ». Dans le cas où l’administrateur aurait oublié ses identifiants, il lui suffira donc d’utiliser cette fonctionnalité pour de nouveau avoir accès au logiciel après une procédure de récupération de compte.

**Déroulement :**

**Scénario initial :** l’administrateur clique sur le bouton « Mot de passe oublié ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant un formulaire de un champ. Prévu pour renseigner l’adresse mail. |
| Remplissage du champ affiché et validation. | 20 |  |
|  | 30 | Vérification effectuée et validée : **#VERA03** |
|  | 40 | Envoi d’un mail contenant un lien pour récupérer le mot de passe. **#MEL01** |

**Scénario secondaire (impossible sans scénario 1) :** L’administrateur clique sur le lien du mail et est renvoyé sur le site web pour créer un nouveau mot de passe (cf. **III. 2) a.**).

### Gestion des groupes

**Description :**

Afin de gérer les droits des employés, les tranches horaires, les accès, ceux-ci seront répartis par groupes. L’administrateur pourra donc, s’il en a les droits, créer un groupe, gérer ses droits, y ajouter des employés etc..

Par défaut, il y a 2 groupes : Groupe Administrateur et le groupe Employés.

* Créer un groupe

Lorsque l’administrateur utilise l’option « Créer un groupe d’utilisateur », cela lui permet de créer un nouveau groupe qui aura ses propres droits et qui pourra contenir un nombre illimité de membre.

**Déroulement :**

**Situation initiale :** L’utilisateur clique sur « Créer un groupe d’utilisateur ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant un formulaire de plusieurs champs. |
| Remplissage du formulaire affiché | 20 |  |
|  | 30 | Vérification effectuée et validée : **#VERB01** |
|  | 40 | Un nouveau groupe est créé, les détails du groupe sont affichés à l’écran **#AFF01** |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 30 | Vérification échouée : #VERB01.  Retour au **point 10** du scénario nominal. |

**Situation finale :** Affichage de la page de détails du groupe créé avec les informations détaillées. Un groupe est créé en base dans la table **groupes**.

* Modifier un groupe

Lorsque l’administrateur utilise l’option « Modifier un groupe », cela lui permettra de modifier les informations générales du groupe, les tranches horaires, les utilisateurs etc…

**Déroulement :**

**Situation initiale :** L’utilisateur clique sur « Modifier un groupe ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant un formulaire de plusieurs champs pré-remplis avec les informations du groupe sélectionné. |
| Remplissage du formulaire affiché | 20 |  |
|  | 30 | Vérification effectuée et validée : **#VERB01** |
|  | 40 | Le groupe est modifié, affichage du groupe modifié à l’écran : **#AFF02**. |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 30 | Vérification échouée : #VERB01.  Retour au **point 10** du scénario nominal. |

**Situation finale :** Affichage de la page de détails du groupe modifié avec les informations détaillées. Le groupe a été modifié dans la table **groupes**.

* Supprimer un groupe

Lorsque l’administrateur est sur la page de gestion d’un groupe, il peut cliquer sur l’option « Supprimer ce groupe ». Ainsi le groupe sera supprimé, et les utilisateurs assignés à ce groupe redeviendront assigné par défaut au groupe **employé** initial.

**Déroulement :**

**Situation initiale :** L’administrateur clique sur « Supprimer un groupe d’utilisateur ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up de confirmation contenant le message : **#MSG01**  l’administrateur peut donc valider ou non la suppression :  Bouton OUI ; Bouton NON ; |
| Clique sur le bouton « OUI » | 20 |  |
|  | 30 | Affichage d’un toast confirmant la suppression **#MSG02**.  Suppression du groupe. |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Clique sur le bouton « NON » | 21 |  |
|  | 22 | Fermeture de la fenêtre pop-up |
|  | 23 | Retour au point 10 du **scénario nominal**. |

**Situation finale :** Affichage du toast de validation. Redirection sur la liste des groupes existants. Suppression du groupe dans la table **groupe.**

* Affecter une tranche horaire à un groupe

Lorsque l’administrateur utilise l’option « Affecter une tranche horaire » sur un groupe d’utilisateur, cela lui permet de définir les heures d’accès et de fonctionnement des badges des utilisateurs de ce groupe.

**Déroulement :**

**Situation initiale :** L’administrateur clique sur « Affecter une tranche horaire » sur un groupe d’utilisateur.

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant la liste des tranches horaires existante ainsi qu’un bouton de création. **#AFF** |
| Sélection d’une tranche horaire | 20 |  |
|  | 30 | Affichage d’une fenêtre Pop-up de confirmation contenant le message : **#MSG02**  L’administrateur peut donc valider ou non l’affectation d’une tranche horaire :  Bouton OUI ; Bouton NON ; |
| Clique sur le bouton « OUI ». | 40 |  |
|  | 50 | Fermeture de la fenêtre pop-up.  Affichage d’un toast confirmant l’affectation de la tranche horaire à un groupe. |

Scénario Alternatif A:

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Clique sur le bouton « Créer une tranche horaire » | 21 |  |
|  | 22 | Cf. E) Créer une tranche horaire  Affectation de la tranche horaire créée si scénario nominal : OK |

Scénario Alternatif B:

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Aucune sélection, fermeture de la fenêtre. | 21 |  |
|  | 22 | Fermeture de la fenêtre pop-up |

Scénario Alternatif C:

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Clique sur « NON ». | 41 |  |
|  | 22 | Retour au point 10 du scénario nominal. |

**Situation finale :** Affichage de la page de détails du groupe modifié. Ajout de l’id de la tranche horaire sélectionnée ou créée.

### Gestion des utilisateurs

**Description :**

L’administrateur peut gérer tous les employés, cela passe par la création, modification, suppression, l’affectation d’un groupe, l’affectation d’un badge etc.

* Créer un utilisateur

L’administrateur peut créer un utilisateur. Cela lui aura pour effet d’ajouter un utilisateur avec toutes les infos nécessaires dans la base de données et d’utiliser les fonctionnalités liées à la gestion des utilisateurs.

**Déroulement :**

**Situation initiale :** L’utilisateur clique sur « Créer un utilisateur ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant un formulaire de plusieurs champs. **#FORMC01** |
| Remplissage du formulaire affiché | 20 |  |
|  | 30 | Vérification effectuée et validée : **#VERC01** |
|  | 40 | Un nouvel utilisateur est créé. Affichage de la page de détail de l’utilisateur. **#AFF** |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 31 | Vérification échouée : #VERB01.  Retour au **point 10** du scénario nominal. |

**Situation finale :** Affichage de la page de détails de l’utilisateur créé. Ajout d’un utilisateur dans la table User.

* Modifier un utilisateur

L’administrateur peut créer un utilisateur. Cela lui aura pour effet d’ajouter un utilisateur avec toutes les infos nécessaires dans la base de données et d’utiliser les fonctionnalités liées à la gestion des utilisateurs.

**Déroulement :**

**Situation initiale :** L’utilisateur clique sur « Modifier un utilisateur ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up contenant un formulaire de plusieurs champs pré-remplis avec les informations de l’utilisateur sélectionné. **#FORMC01** |
| Remplissage du formulaire affiché | 20 |  |
|  | 30 | Vérification effectuée et validée : **#VERC01** |
|  | 40 | L’utilisateur sélectionné est modifié. Affichage des détails de l’utilisateur. **#AFF** |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 31 | Vérification échouée : #VERB01.  Retour au **point 10** du scénario nominal. |

**Situation finale :** Affichage de la page de détails de l’utilisateur modifié. Modification de l’utilisateur en base.

* Supprimer un utilisateur

L’administrateur peut supprimer un utilisateur. Cela aura pour effet de passer l’utilisateur à l’état inactif dans la base de données. Tous ses droits seront donc révoqué, il n’aura donc plus aucun accès.

**Déroulement :**

**Situation initiale :** L’utilisateur clique sur « Supprimer un utilisateur ».

Scénario nominal :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
|  | 10 | Affichage d’une fenêtre Pop-up de confirmation avec le message **#MSG03** et deux boutons :  « OUI »  « NON » |
| Clic sur « OUI ». | 20 |  |
|  | 30 | Affichage d’un toast confirmant la suppression **#MSG02**.  Suppression du groupe. |

Scénario Alternatif :

|  |  |  |
| --- | --- | --- |
| Action | Système | |
| Clic sur « NON ». | 21 |  |
|  | 22 | Fermeture de la fenêtre pop-up. |

**Situation finale :** Fermeture de la fenêtre pop-up. Affichage d’un toast de confirmation de la suppression. Changement de l’état de l’utilisateur à Inactif dans la table **User**.

## Application WEB

# **Modèles de données**

# **Maquettes**