

INFO363 : Analyse et conception par objets TP 1 Diagrammes « Use case »

Jeu Puissance 4

High-Level use case

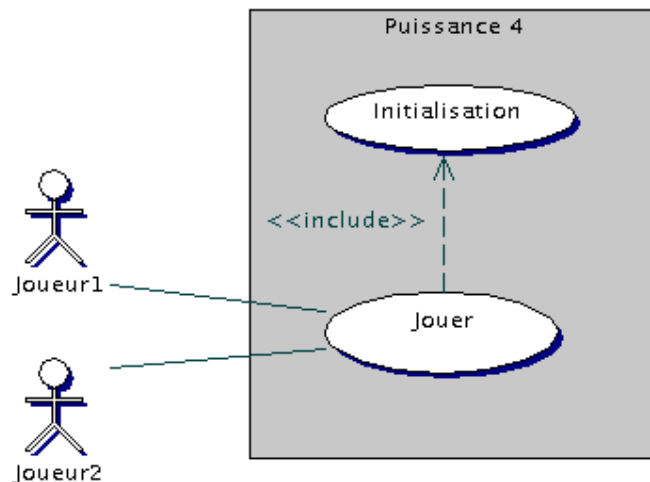
Startup

Un des joueurs lance le programme. Une fenêtre s'affiche contenant une grille de puissance 4 vide. Le programme choisit un joueur au hasard et lui demande de choisir une colonne.

Jouer

Le programme choisit un des joueurs. Il lui demande dans quelle colonne il veut placer ses jetons. Le tableau est mis à jour et les jetons s'affichent en conséquence dans la grille de jeu. Lorsque quatre jetons de même couleur sont alignés, le jeu se termine. Le programme propose une nouvelle partie. Si la table est entièrement remplie sans qu'un joueur n'ait pu aligner 4 jetons de sa couleur, la partie se termine sur un match nul.

Diagramme UML



Detailed use case

Jouer

Acteurs : Joueur 1, Joueur 2

Preconditions : Use Case Startup termine'

<i>Acteur</i>	<i>Système</i>
	Le système désigne le joueur courant
	Il lui demande de positionner sa pièce
Le joueur choisit une colonne et positionne sa pièce	
	Le tableau est mis à jour
	Gagné et fin de partie ?
	On recommence l'ensemble des actions

Exceptions

- Si un joueur clique sur abandon, le jeu doit se terminer.
- Si le joueur désigne une colonne déjà pleine, annuler l'opération et afficher une erreur
- Si l'ensemble des positions sont pleines, le jeu se termine également. C'est un match nul

Gestion de mots de passe

High-Level use case

Startup

L'utilisateur lance le programme et entre sa clé de cryptage. Le système vérifie que la clé est valide, et dans l'affirmative affiche quatre boutons : Recherche, Modification, Suppression et Ajout.

Recherche

L'utilisateur clique sur Recherche puis entre le login de l'utilisateur dont il désire connaître le mot de passe. Si ce login existe, le programme affiche alors le mot de passe en question, sinon il affiche un message "Utilisateur non trouvé".

Modification

L'utilisateur clique sur Modification puis entre un login. Si cet utilisateur existe, le système demande le mot de passe à lui affecté, puis le sauve dans le fichier de base.

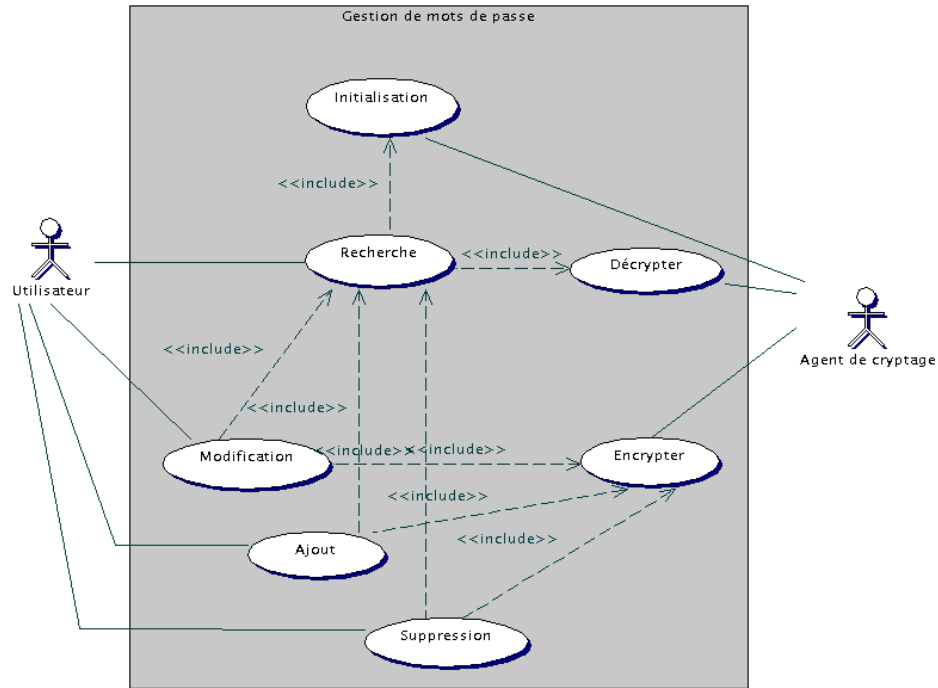
Suppression

L'utilisateur clique sur Suppression puis entre un login. Si celui-ci existe, le système efface l'entrée correspondante du fichier de base, après avoir demandé confirmation à l'utilisateur.

Ajout

L'utilisateur clique sur Ajout puis entre un login et un mot de passe. Si le login n'existe pas encore, le système rajoute une entrée au fichier de base avec les informations entrées.

Diagramme UML



Detailed use case

Startup

Acteurs :Utilisateur, Agent de cryptage

Post-conditions : Le système est lancé, a vérifié que l'accès à l'agent de cryptage était possible et a stocké la clé de cryptage rentrée par l'utilisateur, qui a été certifiée comme valide.

<i>Acteur</i>	<i>Système</i>
L'utilisateur démarre le programme.	Le programme commence par afficher une boîte de dialogue demandant à l'utilisateur d'entrer la clé de cryptage
L'utilisateur rentre sa clef de cryptage dans le champ de texte prévu à cet effet et clique sur OK.	Le système stocke cette clef de cryptage.
	Le système envoie la clef à l'agent de cryptage pour en vérifier la validité.
	Le système reçoit la réponse de l'agent de cryptage
	l'agent de cryptage certifie la validité de la clef : le système affiche l'écran principal, comprenant quatre boutons : Ajout, Suppression, Modification, Recherche

Exceptions

- Si la clef est rejetée par l'agent de cryptage, on demande à l'utilisateur d'introduire à nouveau sa clef.

A tout moment l'utilisateur peut appuyer sur un bouton de sortie du programme

Décrypter

Acteurs : Agent de cryptage

Niveau : sous-fonction

Pre-conditions : Use Case Startup terminé avec succès

Post-conditions : Le système possède en mémoire une version décryptée du fichier de base de données.

<i>Acteur</i>	<i>Système</i>
On envoie à l'agent de cryptage la clef de cryptage qu'il possède en mémoire.	L'agent de cryptage décrypte le fichier de base à l'aide de cette clef et renvoie le fichier décrypté au programme.
Le système stocke le fichier décrypté en mémoire.	

Exceptions

- Si la clef est rejetée par l'agent de cryptage, on demande à l'utilisateur d'introduire à nouveau sa clef.
- A tout moment l'utilisateur peut appuyer sur un bouton de sortie du programme

Recherche

Acteurs : Utilisateur

Niveau : Fonction Système

Pre-conditions : Startup terminé avec succès. Fenêtre principale du programme affichée.

Post-conditions : Le système sait si le login demandé existe dans le fichier de base, et dans l'affirmative il possède une référence vers l'entrée correspondante du fichier. Le fichier de base est décrypté et stocké en mémoire.

Références : Décrypter

<i>Acteur</i>	<i>Système</i>
	Le système affiche une boîte de dialogue demandant à l'utilisateur d'entrer le login qu'il désire rechercher
L'utilisateur rentre le login recherché et clique sur le bouton OK.	Le système obtient une version décryptée du fichier de base. Voir Use Case <i>Décrypter</i> .
	Le système recherche le login dans le fichier de base. Le login est trouvé : le système stocke une référence vers ce login et affiche une boîte de dialogue contenant le login et le mot de passe qui lui est associé.
L'utilisateur confirme les informations affichées en cliquant soit sur Retour au menu principal soit sur Nouvelle recherche. En cas de retour au menu principal : le système ferme la boîte de dialogue et revient à la fenêtre principale du programme.	

Exceptions

- Le login n'est pas trouvé : le système affiche une boîte de dialogue indiquant ce fait
- A tout moment l'utilisateur peut appuyer sur un bouton de sortie du programme

Encrypter

Acteurs :Agent de cryptage

Niveau : Sous-Fonction

Pre-conditions : Startup terminé avec succès. Le programme possède en mémoire une version décryptée et modifiée du fichier de base.

Post-conditions : Cette version modifiée du fichier de base a été réencryptée et stockée par l'agent de crytage.

<i>Système</i>	<i>Agent de cryptage</i>
Le système prend contact avec l'agent de cryptage et lui envoie la clef de cryptage ainsi que la version décryptée du fichier de base.	L'agent de cryptage vérifie la validité de la clé de cryptage ainsi que la structure du fichier décrypté. Il encrypte ensuite le fichier de base et le stocke. Finalement, il renvoie au système un message de confirmation certifiant que l'opération s'est bien déroulée.
Le système affiche une boîte de dialogue indiquant que le fichier de base a été correctement mis à jour.	

Exceptions

- La clé de cryptage n'est pas valide : affichage d'une erreur et demande d'introduction d'une clef valide.
- Le fichier de base est corrompu. Le système affiche un message d'erreur indiquant que les modifications apportées au fichier de base sont perdues.
- A tout moment l'utilisateur peut appuyer sur un bouton de sortie du programme

Les *use-case* **ajout,modification** et **suppression** se déduisent des exemples précédents...