

Java Project

2013

Gestionnaire de Parc Informatique

GROUPE

Alexandre
Ezzat

TEYAR
HAKIM

Contents

Introduction	1
Diagrammes UML	2
Fonctionnalités	5
Interface Graphique	6
Développement	8
Conclusion	9

« à faciliter la gestion de parc informatique »

« simple ne nécessite aucune connaissance particulière en Informatique »

« Ce logiciel est à confier à des personnes responsables, avec de préférence des qualifications en gestion et de confiances »

Introduction

Ce rapport traite d'un gestionnaire de parc informatique implémenté en langage JAVA orienté Objet et disposant d'une interface graphique simple et épurée (voir rubrique Interface Graphique).

Ce logiciel est destiné à faciliter la gestion de parc informatique (voir rubrique Fonctionnalités), il peut par conséquent être utilisé par plusieurs type d'organisations (entreprises, universités, hôpitaux...), son utilisation simple ne nécessite aucune connaissance particulière en Informatique, ce qui rend ce logiciel utilisable par une large population de plus un mode d'emploi est également fournit (voir rubrique Mode d'emploi).

Il n'y a qu'un seul type de profil (celui d'administrateur) et aucune phase d'authentification ce logiciel est donc à confier à des personnes responsables. Ce choix d'une faible sécurité est volontaire de notre part car ce logiciel est à confier à des personnes responsables, avec de préférence des qualifications en gestion et de confiances (secrétaires par exemple).

La langue de ce logiciel est l'Anglais.

« pour donner
une idée
graphique de la
façon dont a été
implémenté ce
logiciel. »

Diagrammes UML

Voici quelques diagrammes UML, pour donner une idée graphique de la façon dont a été implémenté ce logiciel.

Dans un premier temps, il y a un diagramme de classes pour expliciter les objets dégagés de la problématique : « Comment gérer un parc informatique ? ».

Puis un diagramme de cas d'utilisation pour montrer concrètement ce que l'on peut faire avec cet outil-là.

Pour finir avec un diagramme de séquence montrant comment ajouter une salle.

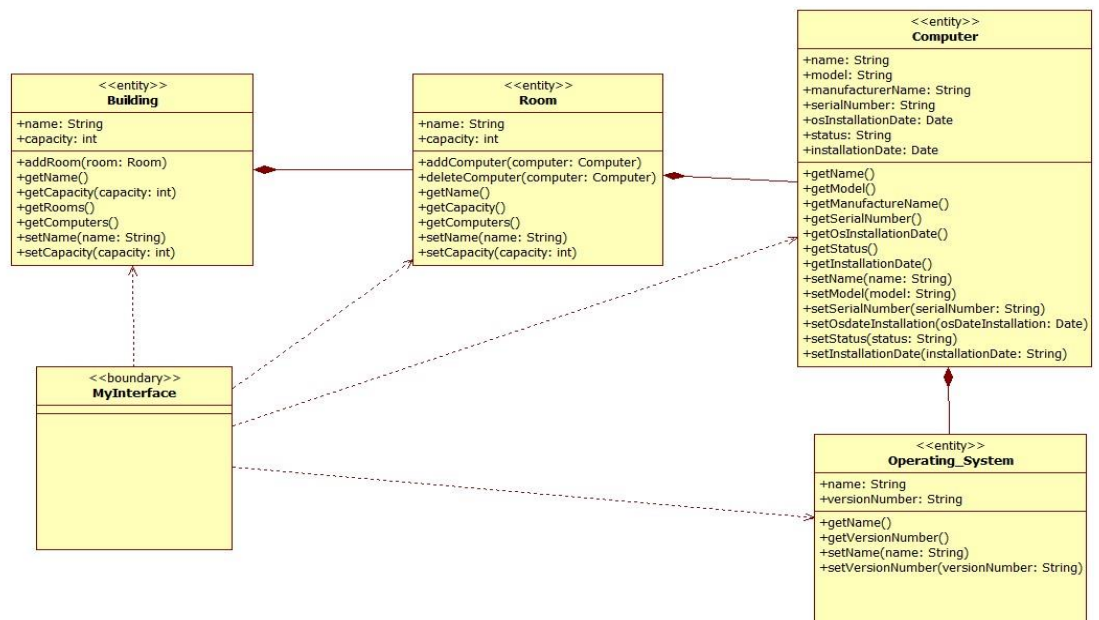


Diagramme de Classes

« d'après nous un parc informatique est composé de bâtiments, de salles et d'ordinateurs les ordinateurs étant composés eux-mêmes d'un système d'exploitation »

Ce diagramme de Classes se justifie par notre vision globale du problème, d'après nous un parc informatique est composé de bâtiments, de salles et d'ordinateurs les ordinateurs étant composés eux-mêmes d'un système d'exploitation, chacune de ces entités disposant d'attributs spécifiques (énoncés dans le sujet de projet). Pour ce qui est des méthodes les seules méthodes nécessaires sont d'ajouter un nouveau bâtiment, ajouter une nouvelle salle, ajouter un nouvel ordinateur ou supprimer un ordinateur, pour tout le reste transfert, mise à jour... les getters et setters de chaque classe étant suffisants.

Un bâtiment ou une salle ne pouvant être supprimé aussi facilement qu'un ordinateur nous avons volontairement décidé d'omettre ces méthodes-là.

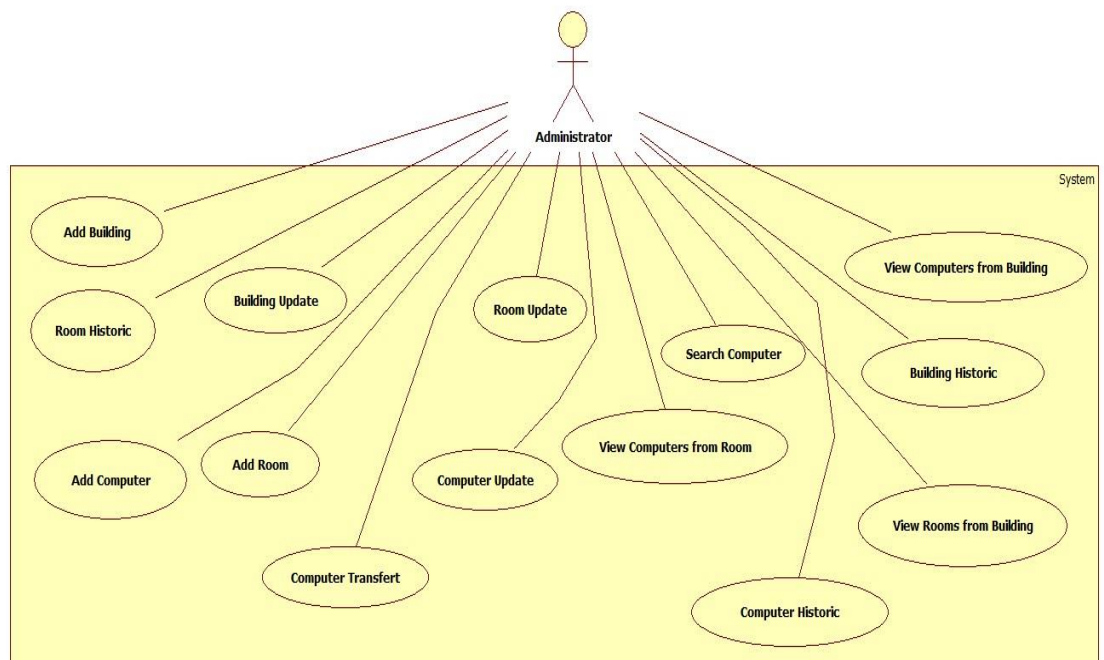


Diagramme de Cas d'Utilisation

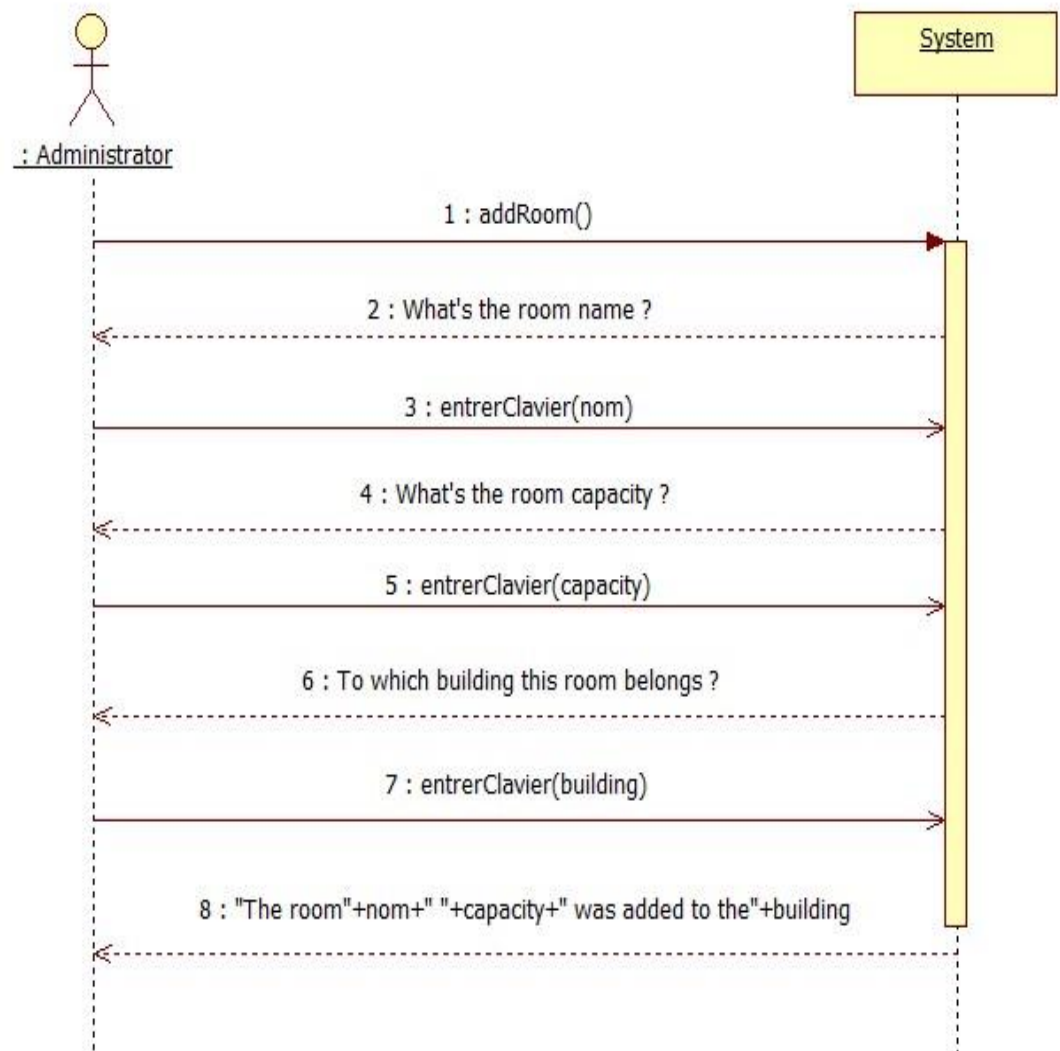


Diagramme de Séquence

Fonctionnalités

Les fonctionnalités proposées par ce logiciel de gestion de parc informatique sont les suivantes (pour avoir une vue graphique de ces fonctionnalités se référer au diagramme de cas d'utilisation) :

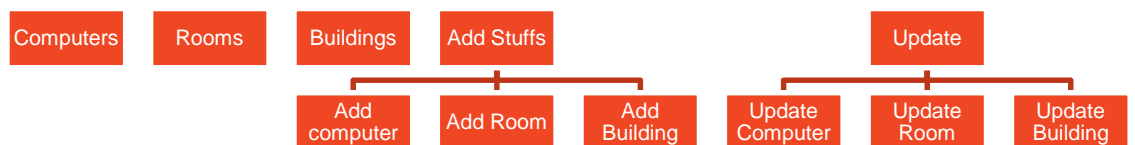
- Ajouter des bâtiments.
- Ajouter des salles aux bâtiments.
- Affecter des ordinateurs à des salles.
- Transférer un ordinateur d'une salle à une autre.
- Avoir la liste des ordinateurs d'un bâtiment.
- Avoir la liste des ordinateurs d'une salle.
- Rechercher un ordinateur
- Mettre à jour les informations d'un bâtiment.
- Mettre à jour les informations d'une salle.
- Mettre à jour les informations d'un ordinateur.
- Afficher l'historique des modifications d'un bâtiment.
- Afficher l'historique des modifications d'une salle.
- Afficher l'historique des modifications d'un ordinateur.

« Ce logiciel est à
confier à des
personnes
responsables »

Interface Graphique

L'interface graphique de ce logiciel est composée de 6 onglets principaux et de sous-onglets, en voici l'arborescence :

- Computers
- Rooms
- Buildings
- Add Stuff
 - Add Computer
 - Add Room
 - Add Building
- Update
 - Update Computer
 - Update Room
 - Update Building



Les onglets Computers, Rooms, Buildings seront consacrés à la recherche respectivement d'ordinateurs, de salles et de bâtiments.

Les onglets Add Computer, Add Room, Add Building seront consacrés à l'ajout respectif d'ordinateurs, de salles et bâtiments.

Les onglets Update Computer, Update Room, Update Building seront consacrés à la mise à jour d'ordinateurs, de salles et de bâtiments.

Ces onglets seront composés de champs texte et de champs de sélection pour récupérer les différents critères de recherche.

De boutons de lancement de recherche, d'actions (filtrer par exemple).

Et de tableaux récapitulatifs ou seront répertoriés les résultats.

Aucune modification de couleur n'a été apportée à l'interface, les thèmes utilisés étant ceux par défaut pour une meilleure compatibilité avec n'importe quel système d'exploitation.

« L'équipe de développement est composée de deux personnes »

« la répartition des tâches (réparti équitablement) »

Développement

L'équipe de développement est composée de deux personnes, les outils de développement utilisés sont StarUML (pour tout ce qui est modélisation), NetBeans et Notepad++ (pour tout ce qui est rédaction de code).

Lors de la conception la priorité a été donné au développement et non pas à l'interface graphique, l'interface graphique ayant été modifiée/travaillée seulement pour répondre aux besoins immédiats.

La planification des différentes étapes du projet ainsi que la répartition des tâches (réparti équitablement) ont été décidé à l'unanimité ainsi, les fonctionnalités à implémenter ont été décidé par le binôme, la modélisation des différents diagrammes ainsi que l'implémentation des méthodes relatives aux objets manipulés a été réalisé par Alexandre, le design du logiciel a été créé par Ezzat, l'intégration des méthodes à l'interface graphique a en grande partie été réalisé par Ezzat et pour finir la documentation a été rédigé par Alexandre.

« nous avons pris plaisir à réaliser ce projet »

« Nous en retirons une expérience positive et profitable. »

Conclusion

Pour conclure ce rapport, nous allons aborder les difficultés que nous avons rencontré lors du développement de cette application JAVA ainsi que ce que nous en avons retiré.

Au niveau des difficultés, la plus notable a été la contrainte de devoir implémenter une interface graphique en effet nous n'avions jamais fait ça auparavant, la manipulation de fichiers (écriture/lecture) pour gérer les historiques et la sauvegarde/chargement a été également un des points délicats de ce développement, pour pallier à ces difficultés-là nous avons dû lire beaucoup de documentation.

Au niveau des enseignements tirés de ces difficultés que nous avons rencontré il y a la capacité à mieux communiquer en équipe (durant toutes les étapes du développement nous sommes restés en contact et avons mis en commun nos travaux et testé leurs compatibilités), la mise en œuvre de nos connaissances « trop théoriques » à un problème concret s'est avéré très utile, et notre capacité à apprendre par nous-même (internet, documentation Java...) a été plus que salutaire.

En somme nous avons pris plaisir à réaliser ce projet, car le sujet nous semblait intéressant, concret et « utile ». Nous en retirons une expérience positive et profitable. Pour ce qui est de l'application à elle-même elle répond actuellement parfaitement aux besoins de gestion de parc informatique, nous pouvons néanmoins toujours améliorer l'interface graphique et/ou ajouter des « fonctionnalités bonus » pour rendre ce logiciel « plus complet » nous aurions pu le faire interagir avec des bases de données en plus de son utilisation de fichiers textes.