



Métro de Bucarest

Document de spécifications

Luiza CICONE
George SAFTA

FILS - 1240F

Table des matières

1	Introduction	2
2	Acteurs	2
3	Exigences	2
3.1	Exigences fonctionnelles	2
3.2	Exigences non-fonctionnelles	3
4	Diagrammes UML	3
4.1	Diagramme de cas d'utilisation	3
4.2	Diagramme de classe	4
4.3	Diagrammes de séquence	5

1 Introduction

Nous avons choisi de développer une application pour le Métro de Bucarest. L'objectif principal est de fournir aux voyageurs un moyen pour calculer le trajet minimal entre deux stations de métro.

2 Acteurs

Pour l'application de métro, les acteurs du système sont les suivants :

- les voyageurs (utilisateurs)
- l'administrateur du SI

3 Exigences

Au vue des différentes techniques comme le brainstorming et le data mining, un certain nombre d'exigences fonctionnelles et non fonctionnelles ont émergé pour cette application.

3.1 Exigences fonctionnelles

1. L'utilisateur peut consulter le plan du métro.

- 1.1 Le système affiche le plan du métro avec tous les stations.
- 1.2 L'utilisateur peut choisir d'afficher les stations sur une seule ligne.

2. L'utilisateur peut trouver un trajet entre deux stations de métro.

- 2.1 L'utilisateur doit introduire 2 stations différentes.
- 2.2 Le système doit calculer le trajet minimal entre les deux stations.
- 2.3 Le système doit sauvegarder la recherche dans un historique pour 48 heures.
- 2.4 Le système peut afficher des trajets alternatifs.

3. L'utilisateur peut s'authentifier

- 3.1 L'utilisateur doit introduire l'identifiant et mot de passe.
- 3.2 Le système vérifie la validité de l'identifiant et du mot de passe.
- 3.3 Une trace de l'authentification sera gardé dans un fichier de log pour 90 jours.

Postcondition : L'utilisateur est authentifié dans le système.

4. L'administrateur peut modifier le plan du métro.

Précondition : L'utilisateur doit être authentifié.

- 4.1 L'utilisateur peut ajouter ou modifier des lignes ou des stations de métro.
- 4.2 L'utilisateur peut ajouter ou changer le nom d'une station.
- 4.3 L'utilisateur peut ajouter une ligne ou modifier une ligne.
- 4.4 Le système vérifie la validité des données : les noms contient que des caractères alphanumériques.
- 4.5 Une trace des modification sera gardé dans un fichier de log pour 90 jours.

Postcondition : La station de métro a changé de nom dans le système.

3.2 Exigences non-fonctionnelles

1. Le système doit fournir un résultat pour la recherche de trajet dans moins de 500 ms.
2. Le système doit fonctionner sur les plate-formes Windows et Unix.
3. Le système doit être fonctionnel tout le temps 7 jours sur 7, 24 heures sur 24.
4. Le système doit être facile à utiliser : l'utilisateur doit accéder à toute opération en moins de 3 actions (cliques).
5. Le système utilise des mots de passe cryptés de 8 caractères.

4 Diagrammes UML

4.1 Diagramme de cas d'utilisation

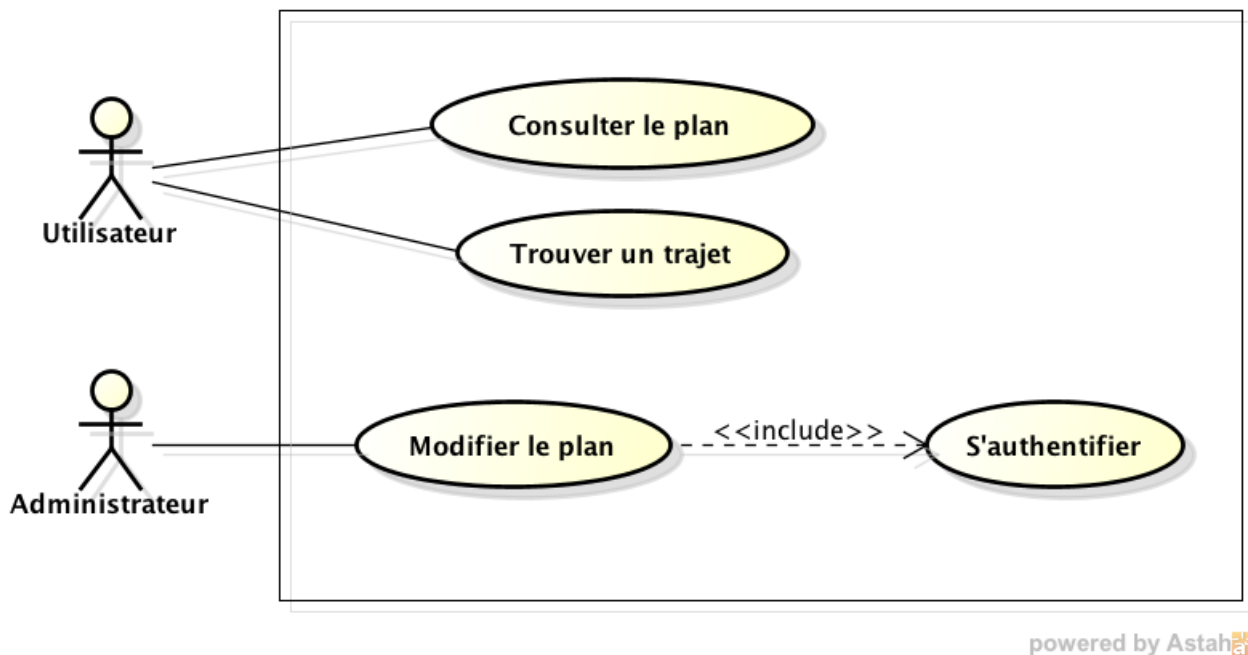
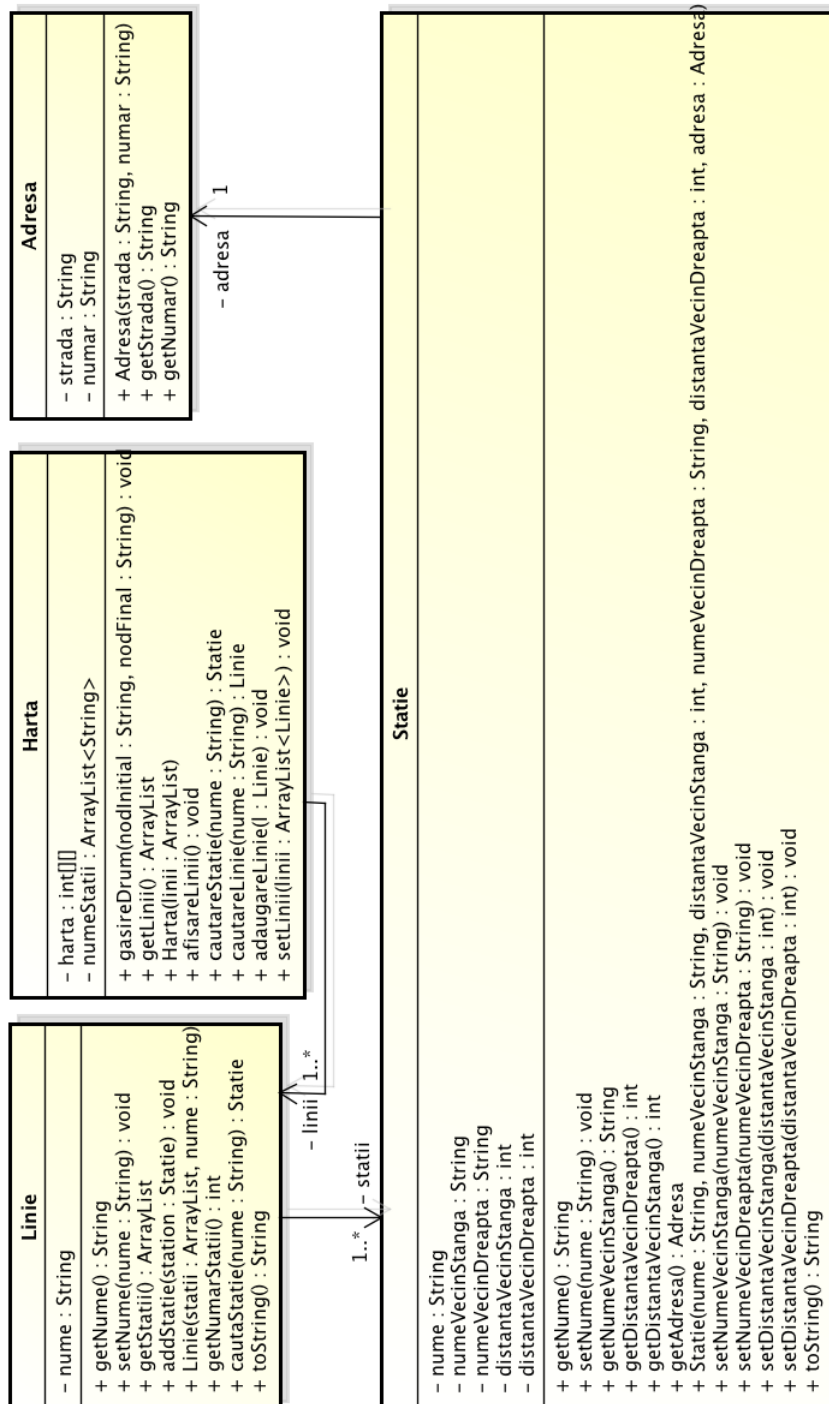


FIGURE 1 – Diagramme de cas d'utilisation

4.2 Diagramme de classe



powered by Astalio

FIGURE 2 – Diagramme de classe

4.3 Diagrammes de séquence

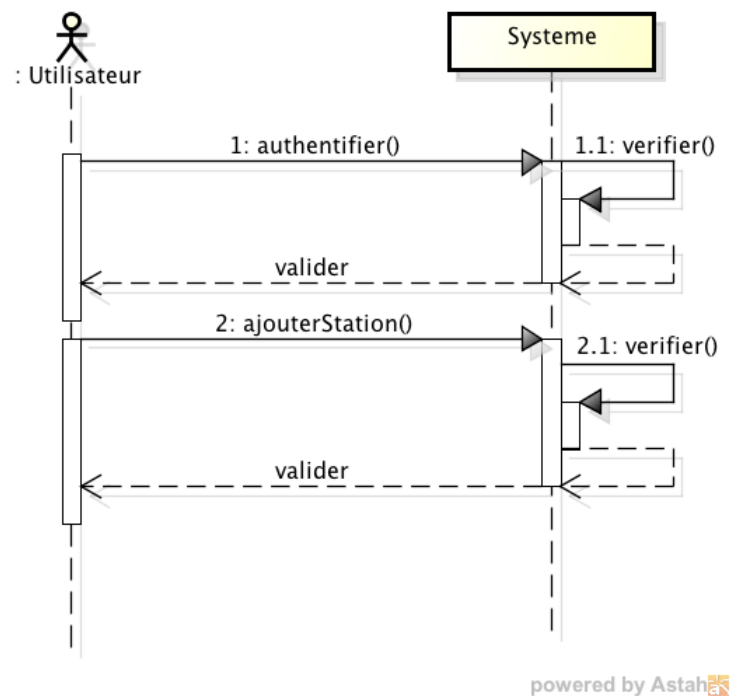


FIGURE 3 – Ajouter une station

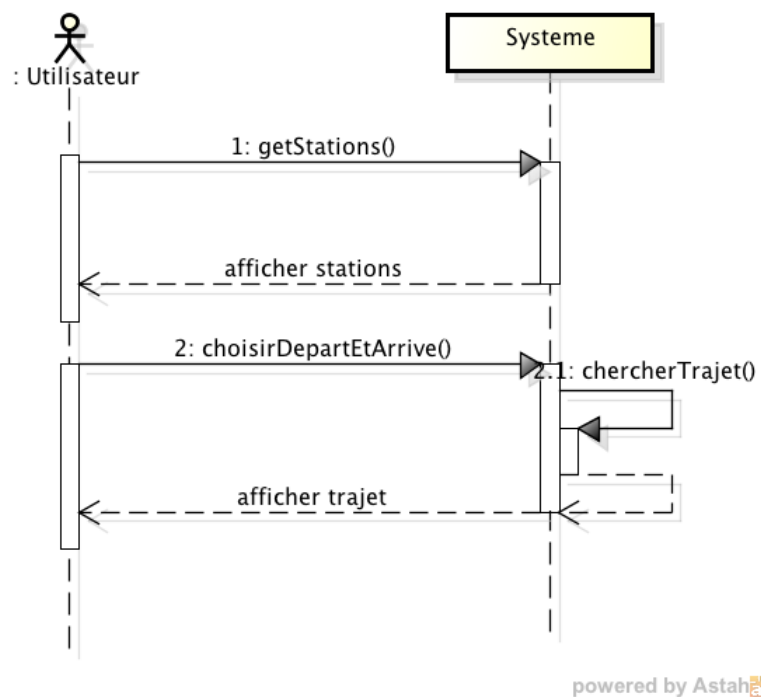


FIGURE 4 – Chercher un trajet