Spacelib

Université Paul Sabatier

Dossier d’analyse et conception

Pierre Liaubet - Fanny MONTUY –Jean-Baptiste Roubelat

2018

Table des matières

[I. Objectifs 3](#_Toc516641623)

[But du projet 3](#_Toc516641624)

[Présentation du sujet 3](#_Toc516641625)

[II. Diagrammes des cas utilisateurs 4](#_Toc516641626)

[III. Diagramme des classes 5](#_Toc516641627)

[IV. Diagrammes Séquences systèmes 10](#_Toc516641628)

[Diagramme Séquences systèmes des UCs Visiteur 10](#_Toc516641629)

[Créer un compte 10](#_Toc516641630)

[S’authentifier 10](#_Toc516641631)

[Consulter Station 11](#_Toc516641632)

[Diagrammes Séquences systèmes des Ucs des Usagers 11](#_Toc516641633)

[Réserver Navette 11](#_Toc516641634)

[Démarrer voyage 11](#_Toc516641635)

[Terminer voyage 12](#_Toc516641636)

[Annuler réservation 12](#_Toc516641637)

[Diagrammes Séquences systèmes des UCs du mécanicien 12](#_Toc516641638)

[S’authentifier 12](#_Toc516641639)

[Lister révisions à faire 12](#_Toc516641640)

[Sélectionner Révision 13](#_Toc516641641)

[Finaliser révision 13](#_Toc516641642)

[Diagramme Séquences systèmes des UCs de l’administrateur 13](#_Toc516641643)

[Ajouter Station 13](#_Toc516641644)

[Ajouter Quai 13](#_Toc516641645)

[Ajouter navette 13](#_Toc516641646)

[Créer mécanicien 13](#_Toc516641647)

[V. Diagramme des composants 13](#_Toc516641648)

[VI. Interfaces des applications 14](#_Toc516641649)

[Application d’administration 14](#_Toc516641650)

[Bornes 14](#_Toc516641651)

[Application des mécaniciens 15](#_Toc516641652)

[Application web pour les usagers 15](#_Toc516641653)

[VII. Jeux de données 16](#_Toc516641654)

[Quais 16](#_Toc516641655)

[Navettes 16](#_Toc516641656)

[Usager 16](#_Toc516641657)

[Station 16](#_Toc516641658)

[VIII. Tests 17](#_Toc516641659)

[Tests de la borne 17](#_Toc516641660)

[Test pas de quai disponible à l’arrivée 17](#_Toc516641661)

[Test pas de navette disponible dans la station de départ 17](#_Toc516641662)

[Test où tout se passe bien 18](#_Toc516641663)

[Table des illustrations 19](#_Toc516641664)

# Objectifs

## But du projet

Ce projet est réalisé dans le cadre du cours d’application d’entreprise. Il a pour objectif de nous évaluer sur nos connaissances et nos compétences concernant la réalisation d’application en JEE.

Tout au long de ce semestre nous avons été formés à concevoir une application en réalisant une analyse d’un sujet en proposant une solution à l’aide de diagrammes. Ce projet a donc également pour objectif d’appliquer

Vous pourrez retrouver l’intégralité de notre projet sur [GitHub](https://github.com/Nexoz/Spacelib).

## Présentation du sujet

Nous devons donc réaliser un service public de transports intergalactiques nommé Spacelib. Il s’agit d’un service de transport gratuits entre stations spatiales permettant à des usagers d’aller d’une station à une autre à l’aide de navette.

Pour ce faire, nous devons développer le système d’information de Spacelib et les différentes applications suivantes :

* Un client lourd permettant aux usagers de réaliser des voyages. Ces clients seront installés sur les bornes présentes dans les stations.
* Un client lourd d’administration. Il permettra de maintenir Spacelib.
* Un client léger pour les mécaniciens qui permet de s’occuper des navettes à réviser.
* Un client léger pour les usagers permettant de voir la carte des différentes stations et de réserver à l’avance un trajet.

# Diagrammes des cas utilisateurs

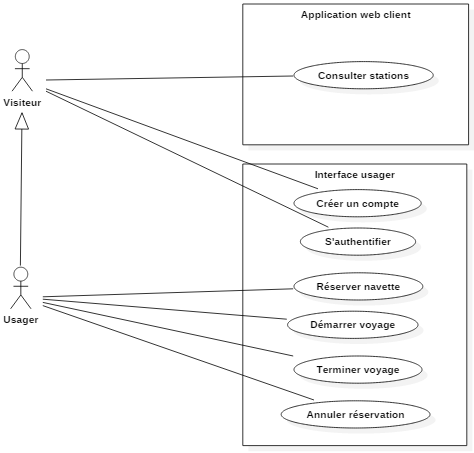


Figure 1 : UC Usager

Pour pouvoir réserver une navette, démarrer un voyage et terminer un voyage, l’utilisateur doit dans un premier s’identifier. Nous sommes partis du principe qu’un usager ne peut pas annuler sa réservation sur une borne car dès qu’il réserve il se dirige vers la navette de sa réservation.

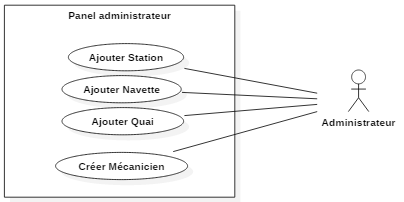


Figure 2 : UC Administrateur

Un administrateur a la possibilité de créer des Stations, des Navettes et des Quais. Cependant pour pouvoir créer une navette, il doit dans un premier temps créer un quai qui va pourvoir l’arrimer. De même pour créer un quai, il faut tout d’abord créer une station.

L’administrateur peut également créer des comptes pour les mécaniciens. C’est à lui de fournir les identifiants du mécanicien.

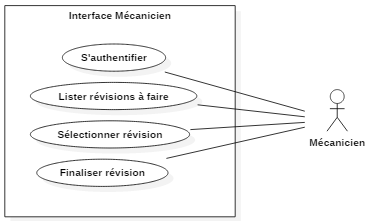


Figure 3 : UC Mécanicien

Le mécanicien avant toute opération, il doit dans un premier temps se connecter. Il pourra ensuite s’occuper des navettes qui en ont besoin.

# Diagramme des classes

Nous avons 10 entités :

* Utilisateur : C’est toutes les personnes ayant accès au système.
* Administrateur : Il s’occupe des stations.
* Usager : Ce sont les personnes qui veulent réaliser des voyages.
* Mécanicien : Ils s’occupent de l’entretien des navettes.
* Opération : Il s’agit d’une action qui a une date de début et une date de fin. Elle se réalise dans un quai et sur une navette. La date d’opération (dateOperation) correspond à la date où l’opération a été créé.
* Révision : Une révision est une opération concernant l’entretien d’une navette. Il se réalise à quai et se fait par un mécanicien.
* Réservation : Une réservation est une opération concernant un voyage d’un usager. Elle possède également une destination. Le nombre de passager du voyage y est enregistré.
* Station : Ce sont toutes les stations que l’administrateur a créées. Elles font partie de Spacelib. Une station est composée de quais.
* Quai : Les quais sont également créés par l’administrateur. Ils sont rattachés à une station. Un quai peut être également réservé pour un voyage.
* Navette : Une navette est arrimée à un quai si elle n’est pas en voyage. Elle possède la liste des opérations qui ont été réalisés sur elle.

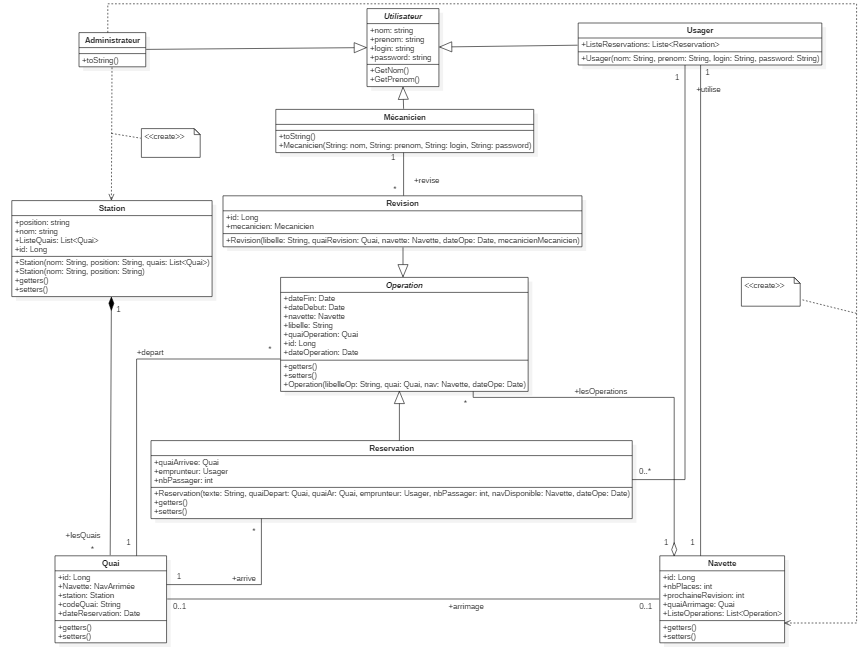


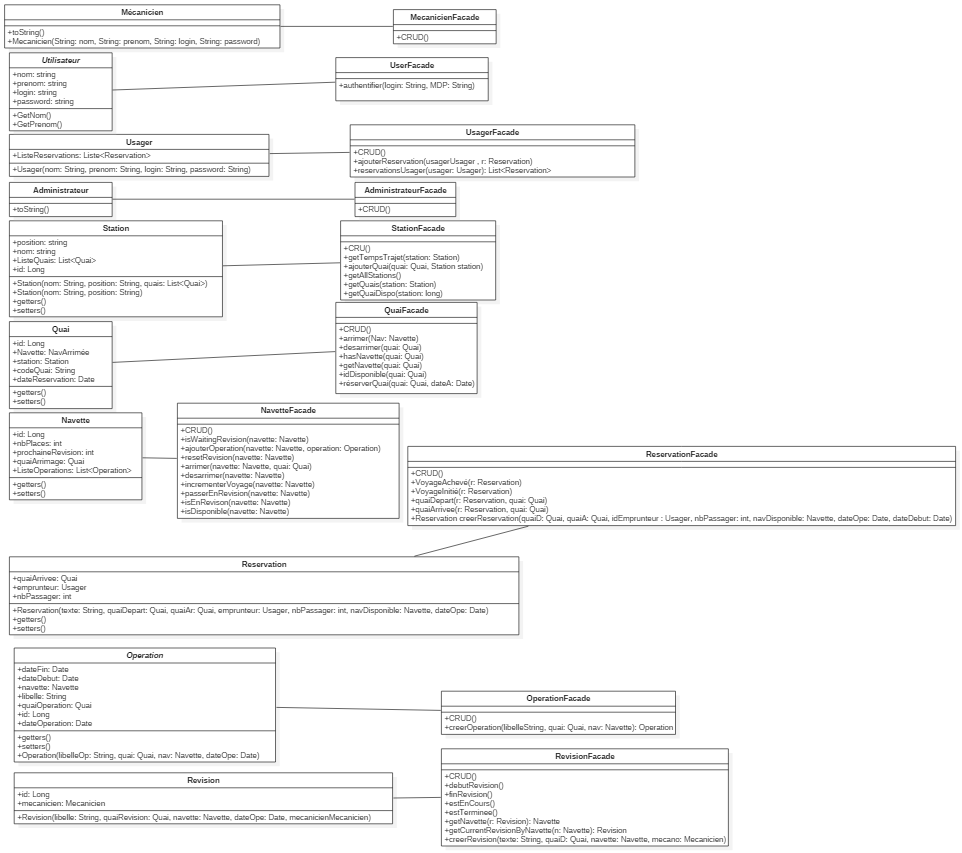
Figure 4 : Diagramme des entités

Pour **la gestion des navettes à réviser**, nous avons ajouté un attribut aux navettes (prochaineRévision). Ce compteur est à 0 si la navette vient d’être révisée. Il est incrémenté à chaque fois que la navette est utilisée lors d’un voyage. Lorsque ce compteur vaut 3, la navette est en attente de révision et n’est plus disponible pour les voyages. Lorsque le mécanicien décide de réviser une navette en attente de révision, le compteur passe à -1. Une fois la révision terminée, il repasse à 0 et la navette est à nouveaux disponible.

Pour **la gestion de la disponibilité des quais** dans la station d’arrivée, nous avons ajouté une date de réservation. Cette date est à la base nulle. Lorsqu’un usager souhaite réserver une navette, nous vérifions si un quai dans la station est disponible. Un quai est disponible s’il n’y a pas de date de réservation et qu’aucune navette y est arrimée. Une fois que l’usager réserve, une date de réservation à la date de l’opération est enregistrée et rend donc le quai indisponible pour éviter des éventuelles collisions entre navettes. Lorsque que le voyage est terminé et que la navette est arrimée, la date de réservation est supprimée.

Toutes les entités ont leur propre façade. Les façades implémentent toutes une interface leur correspondant.

Figure : Diagramme des façades



Les façades sont utilisées dans les différentes gestion. Nous en avons 4 :

* **Gestion Utilisateur** : Cette gestion nous permet de créer des utilisateurs et de pourvoir gérer leur identification.
* **Gestion Stations** : Elle nous permet d’administrer les statons. On peut donc créer des quais, des navettes, des stations.
* **Gestion Révision** : Cette gestion permet de gérer toute la révision dans son ensemble.
* **Gestion Voyage** : Elle nous permet de gérer les voyages des utilisateurs. Elle gère de la création d’une réservation à la finalisation du voyage.

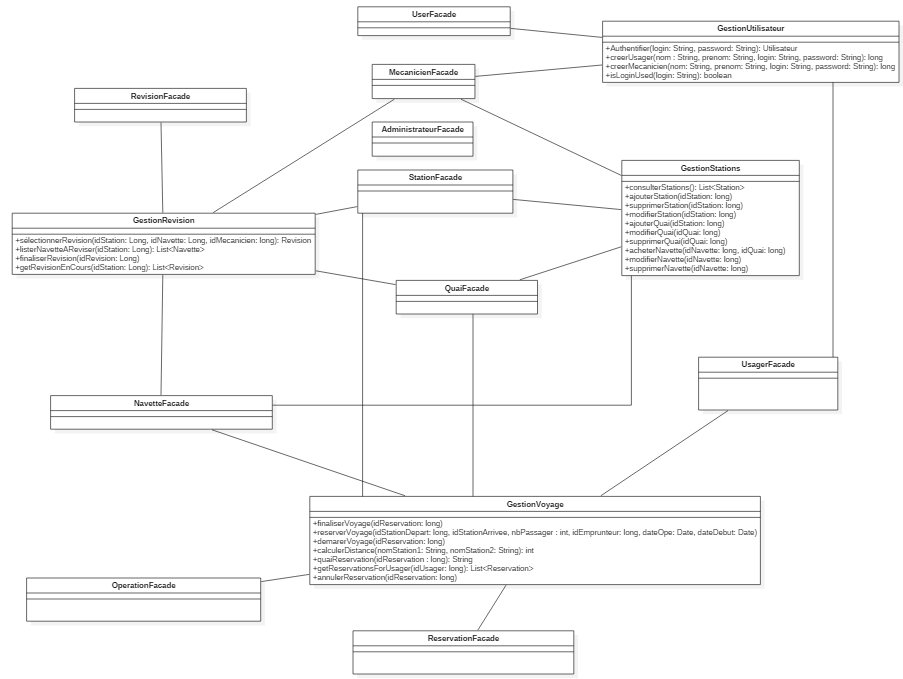


Figure : Diagramme des gestions

Nos applications ont toutes leur propre service. Ces services sont donc adaptés aux comportements des applications.

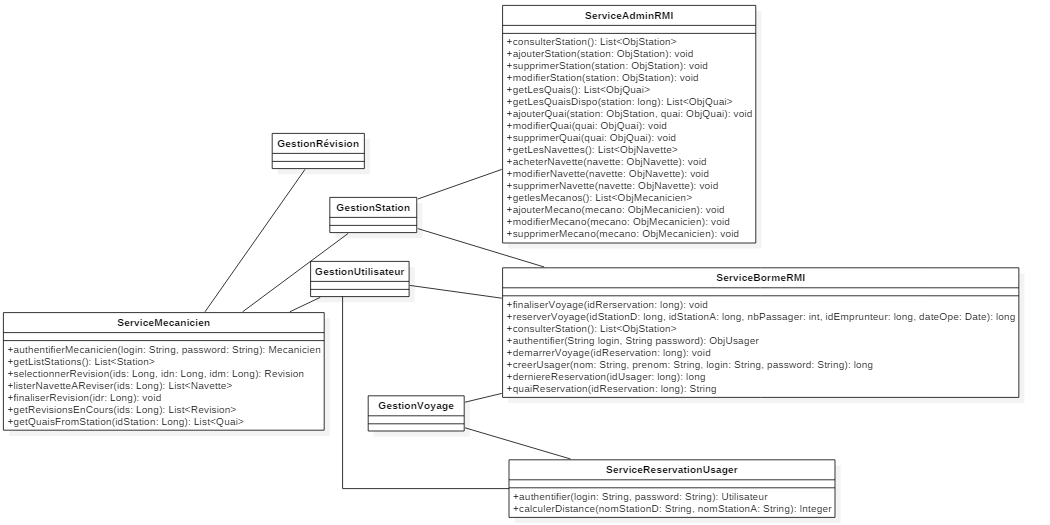


Figure 7 : Diagramme des services

# Diagrammes Séquences systèmes

## Diagramme Séquences systèmes des UCs Visiteur

### Créer un compte

Figure 8 : DSS UC Créer un compte

Si un utilisateur possède déjà un compte avec un login que le visiteur veut, une exception est envoyée. En effet, nous ne pouvons pas avoir dans la base de données des utilisateurs ayant le même login. Cela pourrait poser des problèmes.

### S’authentifier

Figure 9 : DSS UC s'authentifier

Si aucun utilisateur dans la base possède le login ou le mot de passe passé en paramètre, nous récupérons un objet vide.

### Consulter Station

Figure 10 : DSS UC consulter station

## Diagrammes Séquences systèmes des Ucs des Usagers

### Réserver Navette

### Démarrer voyage

Figure 11 : DSS UC Démarrer voyage

### Terminer voyage

Figure 12 : DSS UC terminer voyage

### Annuler réservation

## Diagrammes Séquences systèmes des UCs du mécanicien

### S’authentifier

### Lister révisions à faire

### Sélectionner Révision

### Finaliser révision

## Diagramme Séquences systèmes des UCs de l’administrateur

### Ajouter Station

### Ajouter Quai

### Ajouter navette

### Créer mécanicien

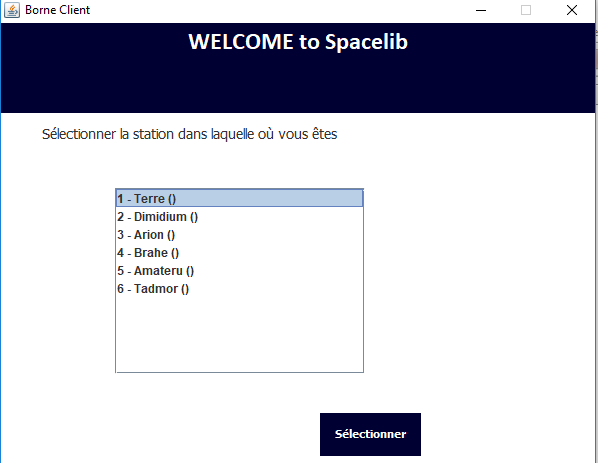
# Diagramme des composants

# Interfaces des applications

## Application d’administration

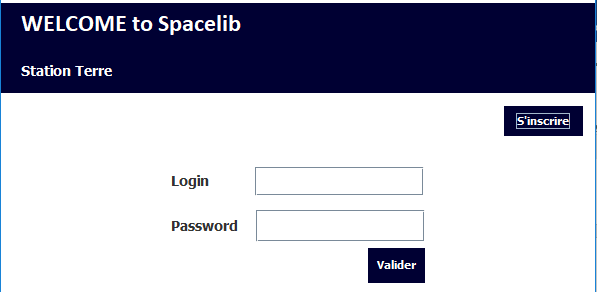
## Bornes

Pour simuler l’emplacement de la borne, nous demandons à chaque utilisation la station où l’on se trouve.

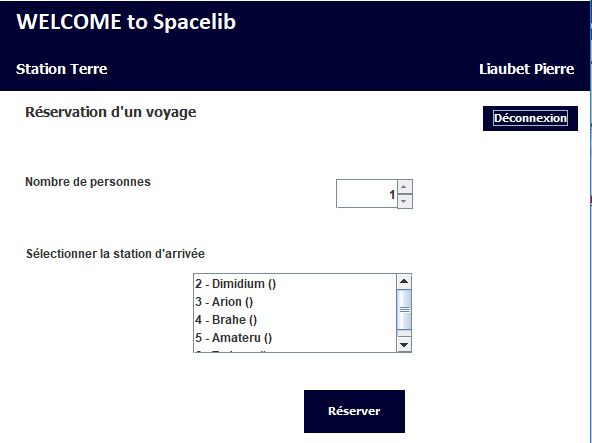


Il suffit de sélectionner la station et de cliquer sur sélectionner.

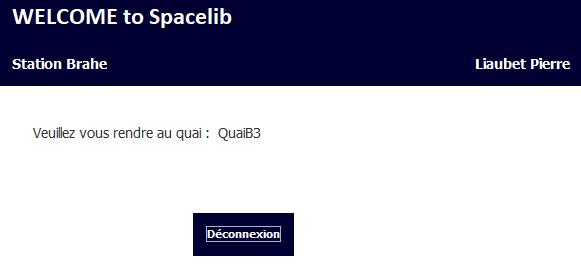
Une fois la simulation nous pouvons soit nous connecter avec notre login et ou notre mot de passe, soit nous pouvons créer un compte.



Lorsque nous sommes connectés sur notre compte, si nous n’avons pas de réservation en cours nous pouvons réserver une navette.



Le nombre de personnes participant au voyage est au minimum de 1 (l’emprunteur). Pour réserver il suffit de sélectionner la station où nous voulons aller et de cliquer sur « Réserver ». Un message nous indique alors le quai où nous devons nous rendre. Il faut se déconnecter pour que le voyage démarre.

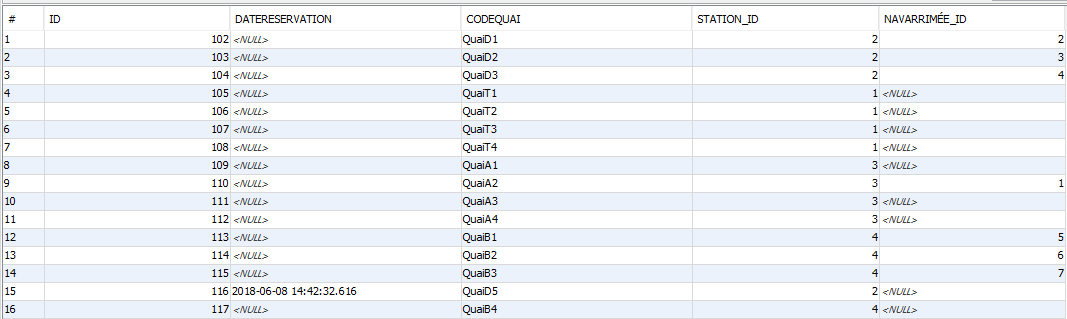


## Application des mécaniciens

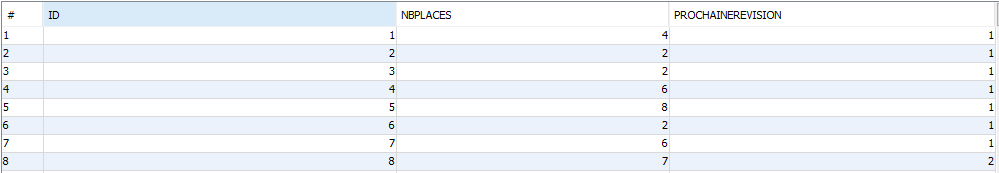
## Application web pour les usagers

# Jeux de données

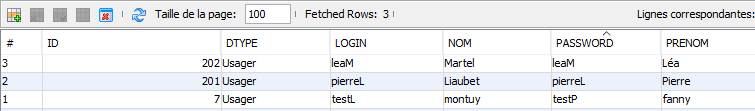
## Quais



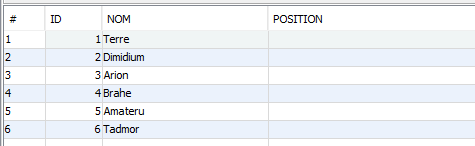
## Navettes



## Usager



## Station

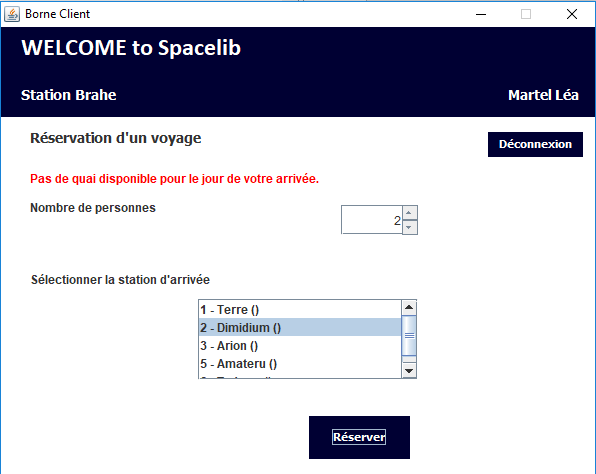


# Tests

## Tests de la borne

### Test pas de quai disponible à l’arrivée

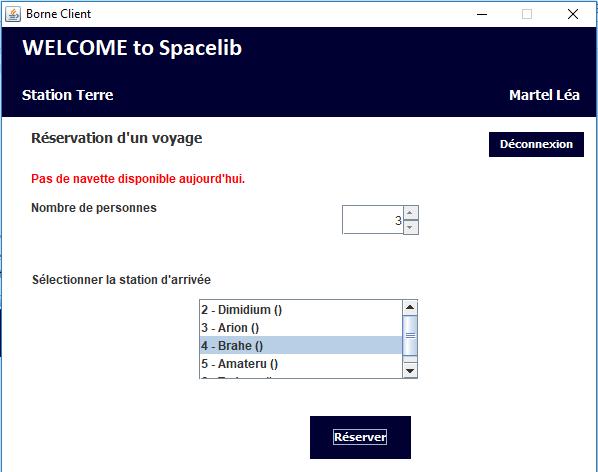
On ne peut réaliser une réservation que si un quai est disponible le jour de l’arrivée. Si aucun quai de la station d’arrivée souhaitée n’est disponible la réservation n’est pas enregistré. Dans le jeu d’essai, la station Dimidium possède des quais dont 3 ont une navette arrimée et le 4ème quai est réservé.



### Test pas de navette disponible dans la station de départ

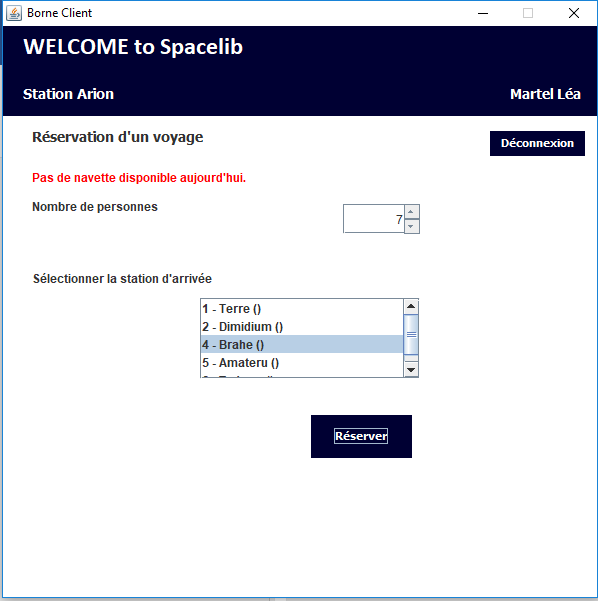
Cas où il n’y a pas de navette dans la station actuellement :

On ne peut pas partir s’il n’y a pas de navette dans la station. Un message prévenant l’utilisateur apparait. Dans le jeu de test il n’y a pas de navette arrimée aux quais de la station Terre.



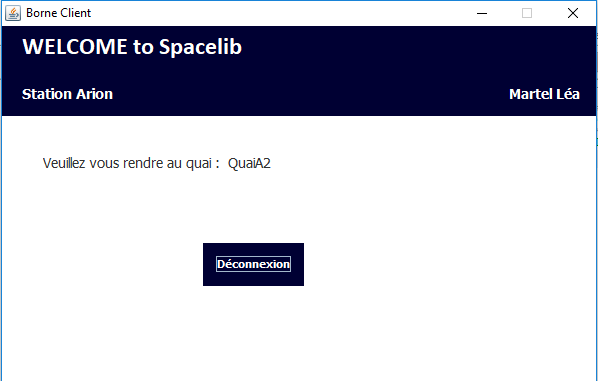
Cas où il n’y a pas de navette avec une capacité suffisante :

Dans la station Arion une seule navette avec une capacité de 4 personnes est arrimée à un quai.



### Test où tout se passe bien

Dans ce test la personne est dans la station Arion où se trouve une navette avec 4 places. L’utilisateur choisit la station Brahe et 3 personnes l’accompagnant.



La déconnexion permet de démarrer le voyage.

La navette est désarrimée du quai.



# Table des illustrations

[Figure 1 : UC Usager 4](#_Toc516613091)

[Figure 2 : UC Administrateur 4](#_Toc516613092)

[Figure 3 : UC Mécanicien 5](#_Toc516613093)

[Figure 4 : Diagramme des entités 6](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613094)

[Figure 5 : Diagramme des services 7](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613095)

[Figure 6 : DSS UC Créer un compte 8](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613096)

[Figure 7 : DSS UC s'authentifier 8](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613097)

[Figure 8 : DSS UC consulter station 9](#_Toc516613098)

[Figure 9 : DSS UC Démarrer voyage 9](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613099)

[Figure 10 : DSS UC terminer voyage 10](file:///D:\Documents\M1\Projets\SpacelibV1\Dossier.docx#_Toc516613100)