



**POLYTECHNIQUE
MONTRÉAL**

LOG1000

Ingénierie logicielle

TP5

Soumis par:
Vincent Samuel-Lafleur
William Younanian
Équipe 411

Le 13 avril 2020

E1) Une mauvaise odeur dans les attributs [/5]

La classe « Appointments » permet de gérer les informations afférentes aux différents rendez-vous. Elle possède un attribut de type « AppointmentsData ».

Examinez la classe « AppointmentsData ».

1) Identifiez le/les nom(s) des odeurs qui se cachent derrière celle-ci et expliquez pourquoi elles sont graves. [/0.5]

L'odeur la plus évidente dans ce cas-ci est qu'il s'agit d'une "data class". Autrement dit, la classe AppointmentData contient simplement des attributs qui pourraient facilement appartenir à la classe Appointment sans rien n'y ajouter de plus.

2) Identifiez le/les nom(s) des restructurations nécessaires pour enlever ce/ces odeur(s) du code. [/0.5]

Les attributs de la classe AppointmentsData pourraient être intégrés à la classe Appointment. Cette dernière serait donc en charge de ses propres données et pourrait effectuer des manipulations pertinentes sur ceux-ci.

3) Identifiez les étapes que vous allez suivre pour restructurer ce/ces odeur(s). Utilisez le même format que le tableau ci-dessous, dans lequel vous devez décomposer la restructuration globale en étapes plus simples. [/1]

Étape	Description
1. Transférer les attributs et méthodes	Prendre l'attribut _data de la classe AppointmentsData et la placer dans la classe Appointment. En faire de même pour les méthodes getData et setData.
2. Faire l'implémentation	Réécrire l'implémentation de ces méthodes dans le fichier Appointment.cpp
3. Ajuster l'implémentation des autres méthodes	Modifier les autres méthodes de la classe Appointment pour prendre en considération son nouvel attribut ainsi que la suppression de la classe AppointmentData.
4. Supprimer l'ancienne classe AppointmentData	Supprimer les fichier cpp et hpp de l'ancienne classe désuète. S'assurer que les dépendances du makefile sont encore à jour.

4) Restructurez le code source et expliquez le raisonnement de vos changements dans le rapport. N'oubliez pas de soumettre les modifications sur Git. [/2]

Comme expliqué précédemment, une "data class" représente une odeur à éviter puisqu'elle nuit à la lisibilité et la réutilisabilité du code. Pour le restructurer, nous avons donc décidé de supprimer la classe AppointmentData qui était une "data class" pour l'intégrer à la classe Appointment et ainsi permettre à cette classe d'effectuer des manipulations directes sur son nouvel attribut, _data. Nous avons donc restructuré le code en suivant les étapes énumérées au numéro 3 avant de tester de nouveau le code pour s'assurer qu'il est toujours fonctionnel.

5) Compilez et essayez de créer un nouveau rendez-vous. Assurez-vous que ce nouveau rendez-vous soit dans la liste des rendez-vous.

Veillez ajouter des captures d'écrans des résultats de ces tests fonctionnels dans le rapport, afin de prouver la validité de vos modifications. [/1]

Ajout d'un rdv :

```
vincent@vincent-ZenBook-UX534FT-UX534FT: ~/Documents/lo...
2. manononon Manon Lepage manon.lepage@ 514-512-1111
213 4e av. Montreal H3T 1P1 5 83604319

Veillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmiere.
2. Prendre rendez-vous avec une infirmiere
3. Voir mes rendez-vous.

[1 ou 2]: 2
Veillez sélectionner une infirmiere par son index: 2

Veillez indiquer la date du rendez-vous [jj/mm/aaaa]: 01/01/2020
Veillez indiquer le moment du rendez-vous [AM/PM]: AM
Veillez indiquer la raison du rendez-vous: gastro

Votre rendez-vous a été correctement enregistré
Votre serez facturé 36 dollars
Appuyez sur ENTER pour continuer...
```

Le rdv dans la liste :

Text Import - [appointments.csv]

Import

Character set: Unicode (UTF-8)

Language: Default - English (Canada)

From row: 1

Separator Options

☐ Fixed width ☒ Separated by

☒ Tab ☒ Comma ☒ Semicolon ☐ Space ☐ Other

☐ Merge delimiters ☐ Trim spaces String delimiter: "

Other Options

☐ Format quoted field as text ☐ Detect special numbers

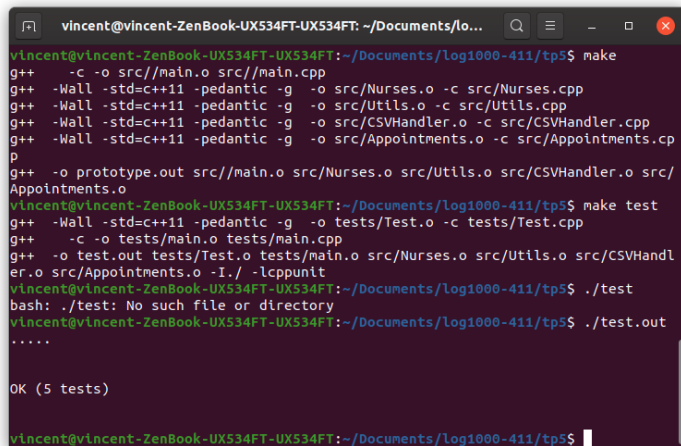
Fields

Column type:

	Standard	Standard	Standard	Standard	Standard
	NurseUserID	PatientUserID	Date	AM/PM	Reason
1	manononon	funkypoly	21/11/2019	AM	Vaccin grippe
2	manononon	jo411	23/04/2019	AM	tousse
3	mclouis	jo411	22/11/2019	PM	grippe
4	mclouis	jo411	29/12/2019	AM	covid-19
5					
6					
7	manononon	jo411	01/01/2020	AM	gastro

Help OK Cancel

Tests automatiques :



```
vincent@vincent-ZenBook-UX534FT-UX534FT: ~/Documents/lo...
vincent@vincent-ZenBook-UX534FT-UX534FT:~/Documents/log1000-411/tp5$ make
g++ -c -o src/main.o src/main.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Nurses.o -c src/Nurses.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Utils.o -c src/Utils.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/CSVHandler.o -c src/CSVHandler.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Appointments.o -c src/Appointments.cpp
g++ -o prototype.out src/main.o src/Nurses.o src/Utils.o src/CSVHandler.o src/Appointments.o
vincent@vincent-ZenBook-UX534FT-UX534FT:~/Documents/log1000-411/tp5$ make test
g++ -Wall -std=c++11 -pedantic -g -o tests/Test.o -c tests/Test.cpp
g++ -c -o tests/main.o tests/main.cpp
g++ -o test.out tests/Test.o tests/main.o src/Nurses.o src/Utils.o src/CSVHandler.o src/Appointments.o -I./ -lcppunit
vincent@vincent-ZenBook-UX534FT-UX534FT:~/Documents/log1000-411/tp5$ ./test
bash: ./test: No such file or directory
vincent@vincent-ZenBook-UX534FT-UX534FT:~/Documents/log1000-411/tp5$ ./test.out
.....

OK (5 tests)

vincent@vincent-ZenBook-UX534FT-UX534FT:~/Documents/log1000-411/tp5$
```

E2) Une mauvaise odeur dans les classes [/4]

Dans la structure actuelle, un « rendez-vous » est représenté comme étant un vecteur de string. La signification associée à chaque index de ce vecteur est définie dans le fichier Constants.hpp.

1) Est-ce la meilleure façon de définir un rendez-vous ? Justifiez votre réponse. [/1.5]

Non, ce n'est pas la meilleure façon de définir un rendez-vous, car c'est difficile de suivre les données du rendez-vous dans le vecteur. En effet, il peut y avoir beaucoup d'éléments dans le vecteur (jusqu'à 10 éléments pour le vecteur des infirmières) et il faut toujours regarder le fichier Constants.hpp pour savoir quelle position du vecteur de string est associée à quelle information. Aussi, cela emploie l'utilisation de beaucoup de constantes qu'on pourrait éviter en faisant autrement.

2) Proposez, sans coder, une restructuration et justifier en quoi celle-ci est pertinente.

Cette restructuration peut-elle s'appliquer dans d'autres fichiers ? [/1.5]

Une solution à ce problème serait de créer une classe Appointment dans laquelle on définirait des variables pour entreposer les données du rendez-vous, ainsi que des méthodes d'accès publiques pour pouvoir lire et modifier le contenu de ces variables. Cette restructuration est pertinente parce qu'elle rendra le code beaucoup plus lisible et claire. Ainsi, dans la classe Appointments (qu'on pourrait renommer, par exemple, par AppointmentManager), au lieu d'avoir un vecteur de vecteurs de string, où chacune de ces vecteurs internes représentent un rendez-vous, on aurait un vecteur d'éléments de type Appointment.

3) Quel effet aura cette restructuration sur le fichier « Constants.hpp ». [1]

Cette restructuration nous permettra de supprimer toutes les lignes #define dans Constants.hpp.

E3) Une mauvaise odeur dans les méthodes [/9]

a) Examinez la méthode « `void Appointments::display()` » /5

1) Identifier le/les nom(s) du/des odeurs qui se cache(nt) derrière celle-ci et expliquez en quoi ce sont des odeurs graves. [/0.5]

-Longueur de la méthode : la méthode est trop longue et fait plusieurs choses (4 étapes).

2) Identifiez le/les nom(s) des restructurations nécessaires pour enlever ce(s) odeur(s) du code. [/0.5]

-Diviser la méthode en plusieurs sous-méthodes

3) Planifiez, étape par étape, comment restructurer ces odeurs, dans le même format que le tableau de l'exercice E1. [/1]

Étape	Description
1. Écrire la déclaration des nouvelles sous-fonctions	Pour réduire la longueur de la fonction <code>display</code> , on crée deux nouvelles sous-fonctions, soit <code>afficherEntete</code> et <code>filtrerAppointments</code> . On déclare ces fonctions dans le <code>.hpp</code> avec leurs paramètres respectives.
2. Écrire l'implémentation de ces sous-fonctions	On réutilise la partie du code de <code>display</code> qui s'occupe d'afficher l'entête du tableau (soit les parties 1 et 2 de la fonction). Pour la fonction <code>filtrer Appointments</code> , on reprend la partie 3 de <code>display</code> .
3. Supprimer les lignes dans la fonction <code>display</code>	Les lignes dans <code>display</code> utilisées par les fonctions <code>afficherEntete</code> et <code>filtrerAppointments</code> peuvent être remplacées par un simple appel à la fonction qui s'occupe de la tâche.
4. Déplacer les déclarations de variables juste avant leur utilisation	Déclarer le vecteur <code>filteredAppointments</code> juste avant l'appel de la fonction <code>filtrerAppointments</code> , et la variable <code>appointmentIdx</code> après l'appel des deux fonctions.

4) Restructurez le code source de cette méthode et expliquez le raisonnement de vos changements dans le rapport. Mentionnez si vous avez créé des nouvelles méthodes ou modifié des méthodes existantes autre que `display`. N'oubliez pas de soumettre les modifications sur Git. [/2]

On a créé deux nouvelles méthodes, soient les méthodes `afficherEntete` et `filtrerAppointments`. Ces méthodes sont privées, car elles seront uniquement appelées par la méthode publique `display`, qu'elles allègent. En effet, avant la restructuration, la fonction `display` s'occupait de plusieurs choses différentes, soient de remplir le vecteur `displayedColumns`, d'afficher l'en-tête du tableau, de filtrer les `Appointments` afin de garder uniquement ceux qui concernent le patient et d'afficher le tableau en entier. C'est assez lourd. Ainsi, on a créé une méthode qui s'occupe de remplir le vecteur `displayedColumns` et d'afficher l'en-tête, puis une méthode qui s'occupe de filtrer les `Appointments`. Cette restructuration diminue grandement le nombre de lignes dans `display`, et elle facilite la compréhension de la méthode.

5) Compilez et exécutez votre logiciel. Veuillez ajouter des captures d'écrans des résultats de tests fonctionnels dans le rapport qui prouve la validité des modifications que vous avez effectués. [/1]

Affichage des rendez-vous :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
[wiyoua@l4712-25 tp5 (master)] $ make clean
rm ./src/main.o ./src/Nurses.o ./src/CSVHandler.o ./src/Appointments.o ./src/Utils.o \
rm ./prototype.out;
[wiyoua@l4712-25 tp5 (master)] $ make
g++ -c -o src/main.o src/main.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/CSVHandler.o -c src/CSVHandler.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Appointments.o -c src/Appointments.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Utils.o -c src/Utils.cpp
g++ -Wall -std=c++11 -pedantic -g -o src/Nurses.o -c src/Nurses.cpp
g++ -o prototype.out src/main.o src/CSVHandler.o src/Appointments.o src/Utils.o src/Nurses.o
[wiyoua@l4712-25 tp5 (master)] $ make run
./prototype.out

Authentification du patient
Veuillez inscrire votre ID d'utilisateur: jo411

Bonjour John Doe!

Voici la liste des infirmières:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis      Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon     Lepage    manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmiere.
2. Prendre rendez-vous avec une infirmiere
3. Voir mes rendez-vous.

[1 ou 2]: 3
Voici vos rendez-vous à venir:

  NurseUserID  PatientUserID  Date  AM/PM  Reason
1.  manononon  jo411  23/04/2019  AM  tousse
2.  mclouis    jo411  22/11/2019  PM  grippe
3.  mclouis    jo411  29/12/2019  AM  covid-19

Appuyez sur ENTER pour continuer...
Menu 19:02
```

Tests :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
[wiyoua@l4712-25 ~] $ cd Bureau/GIT/log1000-411/tp5/
[wiyoua@l4712-25 tp5 (master)] $ make test
g++ -c -o tests/main.o tests/main.cpp
g++ -Wall -std=c++11 -pedantic -g -o tests/Test.o -c tests/Test.cpp
mg++ -o test.out tests/main.o tests/Test.o src/CSVHandler.o src/Appointments.o src/Utils.o src/Nurses.o -I./ -lcppunit
ak[wiyoua@l4712-25 tp5 (master)] $ make run_test
./test.out
.....

OK (5 tests)

[wiyoua@l4712-25 tp5 (master)] $
Menu 19:03
```

b) En examinant la méthode « `int Appointments::getHoursRate()` », vous remarquez probablement certaines odeurs dans le code. /4

1) Identifier les odeurs [/1]

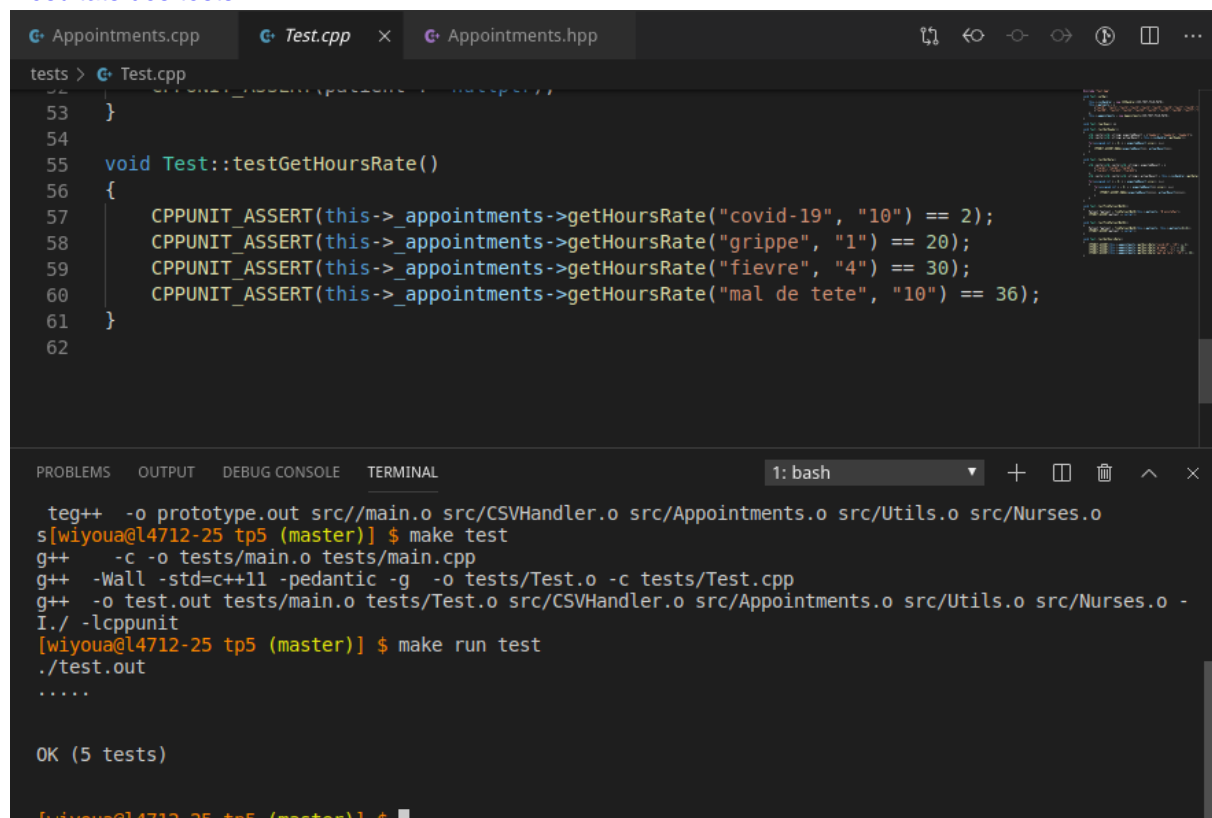
- Nom de la variable `years` pas clair : années de quoi?
- Variable `years` pas déclarée juste avant l'utilisation.
- Condition `if` trop compliquée : Une condition dans le `else` d'une autre condition, dans le `else` d'une autre condition.
- Utilisation d'une condition inutile : `if(years<=2)`. Si cette condition est respectée, le `rate` ne change pas, donc cette condition est inutile.

2) Restructurez le code source de cette méthode et expliquez le raisonnement de vos changements dans le rapport. Mentionnez si vous avez créé des nouvelles méthodes ou modifié des méthodes existantes autre que `getHoursRate()`. N'oubliez pas de soumettre les modifications sur Git. [/2]

- Nom de la variable : `yearsExperience`. Représente le nombre d'années d'expériences de l'infirmière, ce qui est plus représentatif.
- Si `patientDisease` équivaut à "covid-19", on retourne directement le `rate`, soit de 2. Cela nous permet d'éviter de mettre tout le reste du code de la fonction dans un `else`.
- On déclare et initialise `yearsExperience` juste après cette condition, donc juste avant de l'utiliser.
- On enlève la condition inutile pour vérifier `yearsExperience<=2`, et on la remplace par `yearsExperience>2 && yearsExperience<5`.
- On utilise un `else if(yearsExperience>5)` pour changer le `rate` de l'infirmière.

3) Compilez et exécutez votre logiciel. Veuillez ajouter des captures d'écrans des résultats de tests fonctionnels dans le rapport qui prouve la validité des modifications que vous avez effectuées. [/1]

Résultats des tests :



The screenshot shows a C++ IDE with three tabs: `Appointments.cpp`, `Test.cpp`, and `Appointments.hpp`. The `Test.cpp` tab is active, showing the `testGetHoursRate()` function with four `CPPUNIT_ASSERT` tests. Below the code editor, the `TERMINAL` tab is active, displaying the compilation and execution of the tests. The terminal output shows the successful compilation of the program and the execution of the tests, resulting in 'OK (5 tests)'.

```
tests > Test.cpp
53 }
54
55 void Test::testGetHoursRate()
56 {
57     CPPUNIT_ASSERT(this->_appointments->getHoursRate("covid-19", "10") == 2);
58     CPPUNIT_ASSERT(this->_appointments->getHoursRate("grippe", "1") == 20);
59     CPPUNIT_ASSERT(this->_appointments->getHoursRate("fièvre", "4") == 30);
60     CPPUNIT_ASSERT(this->_appointments->getHoursRate("mal de tête", "10") == 36);
61 }
62

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
teg++ -o prototype.out src/main.o src/CSVHandler.o src/Appointments.o src/Utils.o src/Nurses.o
s[wiyoua@l4712-25 tp5 (master)] $ make test
g++ -c -o tests/main.o tests/main.cpp
g++ -Wall -std=c++11 -pedantic -g -o tests/Test.o -c tests/Test.cpp
g++ -o test.out tests/main.o tests/Test.o src/CSVHandler.o src/Appointments.o src/Utils.o src/Nurses.o -I./ -lcppunit
[wiyoua@l4712-25 tp5 (master)] $ make run test
./test.out
.....

OK (5 tests)

[wiyoua@l4712-25 tp5 (master)] $
```

E3) Utilisation des variables [/5]

Afin d'améliorer la compréhension et la lisibilité du code, il est important de minimiser le span, la durée de vie et la portée des différentes variables :

1) Calculez le span moyen, la durée de vie et la portée des variables « choice » et « loggedInUser » dans la méthode « main » dans « main.cpp ». Ne comptez pas les lignes vides. Les accolades et les commentaires comptent comme des lignes. [/0.5]

Variable "choice" :

- Span moyen : 23,67 lignes
- Durée de vie : 73 lignes
- Portée : 79 lignes

Variable "loggedInUser" :

- Span moyen : 11,5 lignes
- Durée de vie : 74 lignes
- Portée : 79 lignes

2) Interprétez les résultats, et trouvez la variable (parmi les deux citées en dessus) qui bénéficiera le plus de la restructuration. Justifiez [/0.5]

La variable choice bénéficiera le plus de la restructuration, car elle est très dispersée dans le main (span très grand). Aussi, comme elle n'est utilisée que quatre fois, il sera facile de rapprocher chaque ligne où elle est utilisée

3) Proposez des restructurations de votre fonction main pour améliorer l'utilisation de cette variable, en utilisant le même format du tableau de l'exercice E1. Notez qu'il faut tout autant minimiser le span, que la durée de vie et la portée. [/1]

Étape	Description
1. Transformer le <code>do{}while();</code> en <code>while(){} </code>	L'utilisation du <code>do..while</code> implique l'utilisation de la variable <code>choice</code> à la fin de la boucle, dans les parenthèses. Si on utilise un <code>while</code> , on peut utiliser la variable au début de la boucle, plus près de là où elle est déclarée et utilisée.
2. Rapprocher l'initialisation de la variable <code>choice</code> de la première fois qu'elle est utilisée	On initialise la variable <code>choice</code> juste avant le <code>while</code> , car c'est là qu'elle est utilisée pour la première fois. Aussi, on change son initialisation à <code>int choice = 1</code> pour que la boucle exécute une première fois.

4) Effectuez cette restructuration dans la méthode « main » et recalculez les nouvelles métriques. N'oubliez pas de soumettre les modifications sur Git.. [/2]

Variable "choice" :

- Span moyen : 3,67 lignes
- Durée de vie : 13 lignes
- Portée : 79 lignes

5) Compilez et testez manuellement les opérations (de l'opération 0 à 3) de la méthode « main ».
Veuillez ajouter des captures d'écrans de vos tests fonctionnels dans le rapport . [1]

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
g++ -Wall -std=c++11 -pedantic -g -o src/Nurses.o -c src/Nurses.cpp
g++ -o prototype.out src//main.o src/CSVHandler.o src/Appointments.o src/Utils.o src/Nurses.o
[wiyoua@14712-25 tp5 (master)] $ make run
./prototype.out

Authentification du patient
Veuillez inscrire votre ID d'utilisateur: jo411

Bonjour John Doe!

Voici la liste des infirmières:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmière.
2. Prendre rendez-vous avec une infirmière
3. Voir mes rendez-vous.

[1 ou 2]: 1

Veuillez sélectionner une infirmière par son index: 1

Il est impossible de prendre rendez-vous avec cette infirmière pour les dates suivantes:
Date          AM/PM
1. 22/11/2019  PM
2. 29/12/2019  AM

Appuyez sur ENTER pour continuer...
Voici la liste des infirmières:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Menu 21:08
```

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
Appuyez sur ENTER pour continuer...
Voici la liste des infirmières:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmière.
2. Prendre rendez-vous avec une infirmière
3. Voir mes rendez-vous.

[1 ou 2]: 2

Veuillez sélectionner une infirmière par son index: 2

Veuillez indiquer la date du rendez-vous [jj/mm/aaaa]: 04/09/2027

Veuillez indiquer le moment du rendez-vous [AM/PM]: AM

Veuillez indiquer la raison du rendez-vous: rona

Votre rendez-vous a été correctement enregistré
Votre serez facturé 36 dollars
Appuyez sur ENTER pour continuer...
Voici la liste des infirmières:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmière.
2. Prendre rendez-vous avec une infirmière
3. Voir mes rendez-vous.

Menu 21:08
```

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
Appuyez sur ENTER pour continuer...
Voici la liste des infirmieres:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmiere.
2. Prendre rendez-vous avec une infirmiere
3. Voir mes rendez-vous.

[1 ou 2]: 3
Voici vos rendez-vous à venir:

  NurseUserID  PatientUserID  Date  AM/PM  Reason
1.  manononon  jo411  23/04/2019  AM  tousse
2.  mclouis    jo411  22/11/2019  PM  grippe
3.  mclouis    jo411  29/12/2019  AM  covid-19
4.  manononon  jo411  04/09/2027  AM  rona

Appuyez sur ENTER pour continuer...
Voici la liste des infirmieres:

  UserID  FirstName  LastName  Email  Telephone  StreetNumber  StreetName  City  PostalCodeYearsExperience  LicenseNumber
1.  mclouis    Louis    Mccartney  louis.mccartn  514-511-1111  1232  3e av.  Montreal  H1T 2V5  2  93202012
2.  manononon  Manon    Lepage  manon.lepage@  514-512-1111  213  4e av.  Montreal  H3T 1P1  5  83604319

Veuillez choisir parmi une des options suivantes:
0. Quitter.
1. Consulter les disponibilités d'une infirmiere.
2. Prendre rendez-vous avec une infirmiere
3. Voir mes rendez-vous.

[1 ou 2]: 0
(wiyoua@l4712-25 tp5 (master)) $
```

E4) Tests [/2]

À la remise, assurez-vous que les commandes make et make test fonctionnent correctement. Assurez-vous que tous les tests unitaires soient valides et effectuez des tests fonctionnels sur le programme, afin de vous assurer que vous n'ayez introduit aucune régression logicielle. Il n'est pas nécessaire de fournir de capture d'écran, mais le tout sera testé lors de la correction et devra être fonctionnel.