

Table des matières

English Summary	3
Introduction.....	8
1. les démarches	9
1.1. Les cours	9
1.2. Le logement	10
1.3. Les Coûts	10
2. Le projet FinanceJ	13
2.1. Introduction.....	13
2.2. Demande client	13
2.3. Conception	14
2.4. Test de boîte noire	16
2.5. Phase de Développement	19
2.6. Résultat	23
2.7. Conclusion	24
3. Enrichissement culturel	26
3.1. Vie Universitaire	26
3.2. Vie quotidienne	26
3.3. Bilan personnel	27
Conclusion	28
Annexes	29

Remerciements

Je voudrais tout d'abord remercier Madame Karine Rouet pour sa disponibilité lors de toute la période de mobilité, ainsi que pour son aide et ses conseils.

Je tiens à exprimer mes sincères remerciements à l'école de technologie supérieure de Montréal et notamment à Madame Elie Anne Goodhue et Monsieur Tanguy Bantas. Qui m'ont aidé à préparer cette mobilité, mais aussi guidé tout au long de l'année.

Enfin, merci à l'IUT de Bordeaux ainsi que l'Université de Bordeaux de permettre la mise en place de tels projets.

English Summary

After three years at the University Institute of Technology, I signed up for an international exchange. As part of a University Diploma in International Technology Studies. This year was held in Montreal I Quebec, province of Canada for a period of eight months. Year in which I validated 48 ECTS.

In this report, we will at first see the steps to take to get there. Seeing course registrations and their validations. Then we will see how I found my home and the characteristics of this one.

Next, I will introduce you the biggest computer science project I did during this year. Presenting the project design, test conception and the development.

Finally, I will make a review of what it has brought me. The cultural enrichment I experienced with University and Everyday Life.

On the way to Canada

My year at ETS started well before the beginning of September. It is necessary to fill many files to go there.

First, in order to get study permit and work permit, people have to obtain the CAQ (certificat d'acceptation du Québec). This paper needs a very solid file which includes evidence about the ability to be housed locally. Evidence of enrollment in a University. Of course, evidence about the identity. And finally, you have to be able to prove your financial capacity to live a year there. For a student in exchange, this amount is more than 10,000 euros.

This is one of the first obstacles to overcome. As far as I'm concerned, I had to go to a bank to get a loan. This process takes a lot of time, and you need to do it well in advance to have time to deal with the rest of the paperwork.

Once the CAQ is obtained, it is now necessary to go to Paris. Only down there is it possible to take the fingerprint. The latter is now mandatory since June 2018. It's still an operation that takes time. Because it is almost mandatory to get an appointment. Waiting time is not even communicated, and I had to wait for almost 2 weeks.

When fingerprinting is complete, the steps for access to Canadian territory are almost complete. However, you still need to get the AVE (Autorisation de voyage électronique). For this one, getting it is fast, and takes very little time.

First Session at the ETS

When I registered for the ETS, I was offered to be enrolled in the beginning of training course. I accepted this proposal, so I had access to Log100 (computer), Mat145 (mathematics), Com100 (communication) and Chem131 (chemistry). All these courses, and all that has been said, was similar to what I was able to experiment with the IUT of Bordeaux. The first part of the volume of a first time is purely theoretical.

As a general rule, the difficulty of the courses for this first session is relatively simple. The

computer course corresponded to the level of the first year of computer science, seen in accelerated. Communication also starts with zeros and does not really require prior knowledge. Mathematics corresponds to a high school level and is not computer mathematics. They are therefore within the reach of all students from DUT. In my case, mathematics was never my forte, but I managed to hang the average.

Second Session

For this session, I enrolled myself in classes and because of this, I had to make choices. Course registration is for a short period of the session. The registration system is therefore saturated as soon as it opens, since the majority of ETS courses have a limited number of places. The places at the courses therefore leave very quickly.

Part of the courses I planned to follow through my pedagogic contract were full. I tried to choose new ones that are close to those initially chosen. So I had to replace ING150 with PHY144 "Introduction to Engineering Physics". But also for a moment, LOG121 by MAT260. For the latter, I finally managed to recover LOG121 "Object Oriented Design", this two weeks after the start of classes. It was a mistake on my part to want to recover this course at all costs. I had trouble catching up on projects.

With regard to the courses the other two courses provided in the pedagogical contract. Namely LOG240 "Tests and maintenance", and MAT210 "Logic and discrete mathematics", these are courses that fit well into the tradition of the computer DUT. LOG240 follows the last semester of DUT. As for MAT210, it takes up and deepened notions also seen in DUT.

Course validation

In Canada, the validation of a course is something very different compared to France. They do not use the "unité d'enseignement" system. Here all the courses are independent. We must therefore validate them individually. Nothing will compensate for a score below average, and then it will be necessary to ironed the course.

In addition, the validation system can change from one course to another, from one teacher to another. But here are the main lines :

The overall average of a student in court does not have to be half (50% or 10/20) to validate it. Some teachers, depending on the class, will raise or lower this threshold. If the class is performing, it can raise the threshold to 55% or even 60%. On the other hand, if the class seems to be in difficulty compared to the teacher's evaluations, it will be able to reduce this threshold to, for example, 45%.

In addition, the ETS requires a minimum average of 50% for written exams in addition to the condition previously discussed. Indeed, as we have seen above, the overall average of a student must correspond to the threshold set by the teacher. It includes all the notes to written assignments, but also notes of practical work and project. The ETS therefore requires that the average of all written exams be greater than or equal to 50%.

There is one exception to this double tier. These are the so-called "off-program" courses. These courses ask for a general average of 60% and nothing else.

Cultural enrichment

University Life

The School of Higher Technology is a very large institute that welcomes a lot of foreign students every semester. The administrative system is pretty well organized, I did not really have a problem.

At the level of the courses, you should know that Quebecers can choose the courses they want to follow during their session from a list of their degree. As a foreign student, I was able to pick something that interested me. The courses are composed of half-hour and half-hour of practical work, which is very similar to the IUT here in France. There are mid-term and end-of-term exams as well as course-based assignments or assignments. The note system is quite complex (as seen in Chapter 1).

Relations with teachers are more flexible than in France. They are open to all questions and available to give help. Some are even called by their first name. The teachers feel close to the students and there is no very formal framework. Everyone knows each other, but it can sometimes be shocking for us French students to see a student from Quebec dissatisfied with a teacher. The courses in their entirety are quite deceptive for foreigners because they are often very complete. We start the first few weeks with simple things but it quickly becomes complex and we must not let go from the beginning because it is difficult to catch up later.

Everyday life

Quebec is known for its rather cold winter. It is necessary to prepare there because it hard a great party of the year (November to April around). The weather always announces two temperatures: the actual temperature, and the temperature felt, which is colder because of the wind factor. Thus, -26° can be felt as -40° . But once you are equipped, the temperature is no longer a problem. And a lot of activities are offered in winter, whether it's the university or the city.

In terms of their rhythm of life, it looks like the rest of the French. The glaring difference is in the evening meal. In Quebec, dinner is taken immediately after the end of the work. Thus, Quebecers eat between 17h and 18h generally.

Public transport is widespread and frequent. The Montreal Transit Corporation (STM) operates 165 bus lines and four metro lines seven days a week. It is possible to take buses twenty-four hours a day. On the other hand, the metros stop around one o'clock in the morning. The shops are open every day, usually until nineteen hours, until four o'clock on the weekends. You will find "convenience stores", small grocery stores, open all around the clock, but priced higher. The city center is full of large shopping centers, part of which is underground to connect them to each other and to the metro.

In terms of food, we are in America so concessions have to be made on certain points. But we find quickly his account. There are no specialties beyond poutine and maple syrup or dishes based on this local product but in general we eat well. Namely that pharmacies do not only sell drugs but also cosmetics and food.

Personal review

I am very positive about my year abroad. Indeed, it is not very easy at first to find oneself far from one's family as well as from one's friends. Nevertheless, I only draw positive from this experience, indeed, I became much more mature than before and much more autonomous. In a general way I would say this stay gave birth in me the desire to travel and to discover the world. I had never been so far and so far from France. I learned to fend for myself, to be a little more curious about myself and to open myself to others. My vision of life has probably evolved over the months. I do not return to France with the same desires and the same expectations but it is a good thing.

Table des matières

Introduction

Après mon DUT informatique, j'ai décidé d'effectuer un DUETI à Montréal au Canada pour une durée de 8 mois. Année au cours de laquelle j'ai validé 48 ects. Je vais, dans ce rapport, aborder les démarches à réaliser pour se rendre sur place, avec le choix des cours, la recherche de logement ou encore les coûts de la vie. Nous verrons ensuite un projet étroitement lié au domaine de l'informatique que j'ai réalisé durant mon séjour. Enfin, nous verrons l'enrichissement culturel que m'a apporté la vie à Montréal, ainsi que le côté vie pratique de cette destination.

1. les démarches

1.1. Les cours

1.1.1. Première session

Lors de mon inscription à l'ETS, il m'a été proposé d'être inscrit automatiquement aux cours de base de début de formation. J'ai accepté cette proposition, et j'ai donc eu accès aux cours Log100 (informatique), Mat145 (mathématiques), Com100 (communication) et Chim131 (chimie). Tous ces cours, et plus généralement tout les cours de l'ETS fonctionnent d'une manière similaire de celle que j'ai pu expérimenter avec l'IUT de Bordeaux. La première moitié du volume horaire d'un cours est purement théorique là ou la deuxième moitié est purement pratique.

En règle générale, la difficulté des cours pour cette première session est relativement simple. Le cours d'informatique correspondait au niveau de la première année de DUT informatique, vu en accéléré. La communication part elle aussi de zéros et ne requiert pas vraiment de connaissances préalables. Les mathématiques correspondent à un niveau lycée et ne sont pas des mathématiques informatiques. Ils sont donc à la portée de tous élève provenant de DUT. Pour mon cas, les mathématiques n'ont jamais était mon fort, j'ai cependant réussi à accrocher la moyenne.

J'ai cependant un avis partagé sur le cours de chimie : ce cours semble s'inscrire dans la continuité des études entrepris par les Québécois. Cependant pour nous, étudiants français, cela remonte à plusieurs années. De ce fait, et dû à mes lacunes dans cette matière, je n'ai pas réussi à valider ce cours.

1.1.2. Deuxième session

Pour cette session, je me suis inscrit moi-même aux cours et par ce fait, j'ai dû faire des choix. Les inscriptions aux cours se font pendant une courte période de la session. Le système d'inscriptions est donc saturé dès son ouverture, car la majorité des cours de l'ETS ont un nombre de places limité. Les places aux cours partent donc très vite.

Une partie des cours que j'avais prévu de suivre via mon contrat pédagogique étaient donc plein. J'ai essayé d'en choisir de nouveaux qui se rapprochent de ceux initialement choisis. J'ai donc dû remplacer ING150 par PHY144 "Introduction à la physique du génie". Mais aussi pendant un moment, LOG121 par MAT260. Pour ce dernier, j'ai fini par réussir à récupérer LOG121 "Conception orientée objet", cela deux semaines après le début des cours. Ce fut une erreur de ma part de vouloir à tout prix récupérer ce cours. J'ai en effet eu du mal à rattraper mon retard sur les projets.

En ce qui concerne les deux autres cours prévus dans le contrat pédagogique. À savoir LOG240 "Tests et maintenance ", et MAT210 "Logique et mathématiques discrètes", ce sont des cours qui s'inscrivent bien dans la lignée du DUT informatique. LOG240 fait suite au dernier semestre de DUT. Quant à MAT210, il reprend et approfondie des notions aussi vues en DUT.

À la fin de ce semestre, j'ai validé : MAT210, LOG240 et PH144.

1.1.3. La validation des cours

La quasi-intégralité des cours de l'ETS se valide par un système de double palier. En effet, il ne suffit pas d'avoir juste la moyenne dans le cours pour valider celui-ci.

Tout d'abord, la moyenne général d'un élève au cours ne doit pas forcément être la moitié (50 % ou 10/20) pour valider celui-ci. Certains professeurs, en fonction de la classe, vont faire monter ou descendre ce seuil. Si la classe est performante, il peut faire monter le seuil à 55 % voir même à 60 %. À l'inverse, si la classe semble en difficulté par rapport aux évaluations du professeur, il pourra faire reculer ce seuil à par exemple 45 %.

Qui plus est, la moyenne aux contrôles écrits est aussi à obtenir. En effet comme nous l'avons vu précédemment, la moyenne général d'un élève doit correspondre au seuil fixé par le professeur. Elle regroupe l'ensemble des notes aux devoirs écrits, mais aussi les notes de travaux pratiques et de projet. L'ETS exige donc que la moyenne de tout les examens écrits soit elle supérieur ou égal à 50 %.

Il y a cependant une exception à ce double palier. Ce sont les cours dits "hors programme". Ces cours-ci demandant d'obtenir une moyenne général de 60 % et rien d'autres.

1.2. Le logement

1.2.1. La recherche

En général, les étudiants étrangers habitent en colocation, c'est pourquoi j'ai commencé par prospecter par ce type d'annonce. Connaissant plusieurs autres personnes qui partaient vivre à Montréal au même moment, j'ai fini par penser que cela serait chose aisée. En parallèle j'ai tout de même candidaté aux résidences étudiante de l'ETS.

La recherche la collocation a fini par prendre beaucoup de temps, sans pour autant mener à une solution. Et au mois de juillet, j'ai appris que j'avais accès aux résidences l'ETS. J'ai fini par choisir cette option pour clore ma phase de recherche de logement.

1.2.2. L'appartement

J'ai donc habité dans un appartement 5 1/2 qui équivaut un T5 en France au prix de 480 \$ par mois (environ 320 €). L'appartement est situé à Griffintown et, est composé de 4 chambres, de deux salles de bain et d'une grande pièce principale composé d'un salon et de la cuisine. Il est situé à 2 min de l'ETS, ce qui est un grand plus. Le prix du loyer est relativement cher (4 étudiants qui payent ~500 \$ soit ~2000 \$ par mois) par rapport au prix des autres logements de cette superficie dans ce quartier. Une partie du prix se justifiant par la proximité de l'école. Ce qui permet de ne pas forcément payer un abonnement aux transports en commun, qui est relativement cher. Il n'y avait aucune caution à payer ni assurance à fournir, et les charges sont incluses dans le prix.

1.3. Les Coûts

1.3.1. Avant le départ

Il faut savoir qu'une grosse partie des dépenses que j'ai effectuées dans cette année se trouvent avant mon départ. Parmi ces dépenses préalables au départ, il y a de nombreuses dépenses en ce qui concerne les frais de dossier : pour la demande de permis de travail, de permis d'études ou encore pour l'inscription à l'ETS. Pour chacun de ces dossiers, il faut compter en moyenne 40 euros par dossier.

Il faut savoir qu'une bonne partie de ces dossiers sont fait pour obtenir un permis d'étude et un permis de travail. Ce qui est important à savoir et que, peut importe les frais de dossier, l'obtention de ces permis exige une preuve de capacité financière plutôt énorme. J'ai en effet vu mes dossiers refusés une première fois, malgré un emprunt à la banque pour une solvabilité montant à 10 000 euros. En effet, le site de l'immigration du Canada réclame environ 10 000 euros de preuve pour obtenir des permis. Pour ma part, ma preuve montait à 10 000 environ. Mais puisque qu'elle ne les dépassait pas franchement, la demande a été refusée. Contraignant à de nouveau trouver des fonds, et à effectuer encore une demande.

Concernant les permis, depuis juillet 2018, il faut maintenant fournir ses empreintes digitales pour obtenir ces derniers. Il faut donc compter un allé retour à Paris, car il n'y a qu'un seul bureau en France et il se trouve à Paris.

1.3.2. A Montréal

Premièrement, il faut faire attention avec sa carte de paiement. Il a évidemment des coûts liés aux frais à l'étranger lors de retraits ou de paiements par carte. Pour résoudre ce problème, je n'ai pas choisi d'ouvrir un compte sur place, en raison de la durée de mon séjour, mais plutôt de prendre une option pour ma carte adaptée à mon type de voyage auprès de ma banque.

La plus grande des contraintes que j'ai pu rencontrer concernant les paiements sur place est que bon nombre de magasins ou organisations n'accepte pas le paiement par carte de crédit, mais seulement par carte de débit. Par exemple, pour payer mon loyer, j'étais contraint de retirer les sous en espèces pour les donner au bureau des résidences sachant que les sommes s'élevaient à environ 500 \$ le premier semestre, cela était loin d'être pratique.

En ce qui concerne la vie de tous les jours, le coût de la vie s'apparente au coût de la vie dans une grande ville française. Comme présenté un peu plus tôt, le logement est d'environ 320 euros par mois. À cela, il faut rajouter le prix de l'alimentation. Qui, en ce qui me concerne, me coûtait entre 20 et 35 euros par semaine, soit environ 100 euros par mois.

1.3.3. Santé

Dès mon inscription à l'ETS, je devais transmettre un formulaire rempli par la CPAM de Bordeaux indiquant les dates d'arrivée et de départ du Québec. Les démarches furent assez simples, car la France bénéficie d'ententes avec le Québec sur le régime de santé. Une fois arrivé à Montréal, j'ai pu obtenir ma carte me rattachant à la régie de l'assurance-maladie du Québec (RAMQ).

Grâce à cette entente, il suffit de payer son assurance-maladie en France, et l'on profite des services santé du Québec comme si l'on paierait l'assurance-maladie du Québec.

Je n'ai pas beaucoup plus d'informations à ce sujet, étant donné que je n'ai jamais été malade durant mon séjour. J'ai cependant pu noter que le dentiste n'est pas rattaché au régime de l'assurance-maladie. Les cabinets sont donc tenu par des entreprises privées, et le coût résultant est extrêmement cher comparé aux prix en France.

1.3.4. Télécommunications

La téléphonie au Canada est très peu développée, et le prix des services fournis est extrêmement élevé. Surtout en ce qui concerne le prix des données permettant l'accès à Internet. En moyenne 90 dollars pour appel et sms illimité plus un giga de données internet. J'ai donc choisi de ne pas souscrire à un forfait téléphonique. À savoir que Montréal est une ville qui dispose d'un très bon réseau de wifi publique. Grâce à cela, j'ai pu effectuer mes 8 mois sans trop de problèmes d'un point de vue télécommunication.

2. Le projet FinanceJ

Durant la seconde session, j'ai pu travailler sur un projet de maintenance informatique, grâce au cours LOG240 "Tests et maintenance".



Je vais ici m'attarder sur ce que j'ai pu découvrir comme méthodes et outils de conception et de développement. Je ne vais pas pour autant négliger ce que j'ai déjà pu voir à l'IUT, mais je pense qu'il est plus pertinent de vous présenter ce que j'ai pu apprendre.

2.1. Introduction

La firme de consultation MaintenancePlus, qui est un spécialiste dans la maintenance et le test de logiciels, vient d'assigner à mon équipe son premier contrat de travail. La situation de l'entreprise BudgetPersonnel2000, pour laquelle MaintenancePlus requiert vos services, est la suivante :

- BudgetPersonnel2000 est une entreprise qui vend des services à des particuliers afin de les aider à gérer leur budget.
- Il y a un an, l'un de ses gestionnaires a eu l'idée de développer un logiciel que l'entreprise pourrait vendre à ses clients pendant les rencontres, afin de générer plus de bénéfices.
- Une tierce personne a donc été embauchée par BudgetPersonnel2000 pour développer et assurer la maintenance de son logiciel, qu'elle a dénommé FinanceJ.
- Ayant trouvé un autre emploi, cette tierce personne vient de quitter son poste à BudgetPersonnel2000 qui se presse alors de recueillir les services de la firme MaintenancePlus pour assurer désormais la maintenance de FinanceJ.

L'état du logiciel FinanceJ n'est pas reluisant. Ses utilisateurs trouvent qu'il n'est pas assez convivial, qu'il exécute des opérations qui peuvent briser l'intégrité de la base de données, et que certaines de ses fonctionnalités ne s'exécutent même pas. L'équipe de développement a donc la mission de prendre en charge le logiciel dans un environnement de maintenance réel pour pouvoir l'améliorer, conformément aux normes qui régissent le domaine.

[.jpg]

Le but du projet est donc de reprendre cette application, et de l'améliorer pour l'emmener vers une version utilisable.

Pour cela, une grande phase de conception est nécessaire. Avec dans l'ordre : l'analyse de la demande client. Une conception des cas d'utilisation qui en découlent. La conception des tests à implémenter pour définir si les modifications sont bonnes. Et enfin le développement qui commencera par la mise en place d'outils Maven, puis par l'implémentation en Java EE.

2.2. Demande client

Le projet commence par une phase une demande client. Il est ainsi demandé à l'équipe de développement d'apporter des modifications apportant de nouvelles fonctionnalités à l'application.

Ces modifications devant apporter “un système de droit d'accès au logiciel (login, mot de passe, accès avec différent droit)”.

En effet, lorsque le programme est exécuté, l'application se lance directement. Il n'y a aucune sécurité, et tous les employés de l'entreprise ont les mêmes droits dessus. Cela doit donc changer, avec une connexion sécurisée qui permettra de limiter les droits en fonctions des employés.

2.3. Conception

Maintenant que la demande du client a été explicité. Nous nous sommes tourné vers l'analyse de celle-ci. La demande est assez simple à première vu, mais elle entraîne plusieurs modifications du fonctionnement du logiciel. Nous avons donc travaillé dessus et nous sommes arrivés à la conclusion que cette demande comprends trois fonctionnalités sous-jacentes. Et chacune de ces fonctionnalités comprend des sous-fonctionnalités. Les vois-ci :

Un système d'authentification (fenêtre prévue à cet effet) comprenant :

- Une création d'utilisateurs (login + password).
- La correspondance de la pair login password avec database pour se connecter à l'application.
- La suppression d'utilisateur.

Un Système d'attribution de droits (fenêtre prévue à cet effet) :

- Master user capable de donner des droits aux autres users.
- Master user capable de supprimer des droits aux autres users.

Limitation du logiciel en fonction des droits (partie non visible des modifications) :

- Accès à certaines parties du logiciel en fonction des droits.
- Accès à certaines fonctionnalités en fonction des droits (tout le monde ne doit pas pouvoir supprimer des utilisateurs / transaction / etc...).

Maintenant que les fonctionnalités majeurs à ajouter dans le programme ont été définies, tournons nous vers les nouvelles manières d'utiliser le logiciel lorsque que les ajouts auront été implémenté.

2.3.1. Cas d'utilisation

Pour cela, nous avons tout d'abord établie plusieurs cas d'utilisation. Ces cas d'utilisations vont permettre de définir les principales manières d'utiliser le logiciel. Ils contiennent chacun quatre parties.

La première étant les préconditions, on regroupe ici toutes les conditions qui permettent à l'utilisateur d'effectuer l'action voulue. effectuer l'action voulue.

Exemple : Ici, nous allons prendre l'exemple de l'ajout d'un compte dans la base. Compte qui comprend un nom et une description. L'utilisateur doit donc être connecté au logiciel et dans l'onglet d'ajout.

La deuxième partie regroupe les post-conditions. Ici, ce sont toutes les conditions qui permettent

d'affirmer que l'utilisation voulue s'est bien déroulée. Par exemple, si l'on reprend l'exemple de l'ajout du compte dans la base, une post-condition serait qu'un nouveau compte a bien été ajouté dans la base.

La troisième partie est composée du scénario principal. Celui-ci regroupe l'ensemble des actions utilisateur qui compose l'utilisation voulue. Toujours avec l'exemple de l'ajout du compte dans la base. On va par exemple s'attendre que l'utilisateur remplissent les champs comme le nom et la description, ou encore qu'il appuie sur le bouton d'ajout.

Enfin, la dernière partie regroupe les scénarios alternatifs. Comme pour le scénario principal, c'est une suite d'actions qui composent une utilisation. Mais ici, ce n'est pas l'utilisation principale de la fonction logicielle traitée. Généralement, on regroupe ici les cas qui peuvent amener à une erreur (mais pas uniquement). Avec l'exemple de l'ajout du compte dans la base. Un scénario alternatif serait de tenter d'ajouter un compte dans la base sans avoir renseigné le nom, ce qui va mener à un message d'erreur.

Le nombre de cas d'utilisation que nous avons conçus est de 3.

Maintenant que la définition des cas d'utilisation est clair, voici un extrait de nos cas d'utilisation. Il y décrit l'utilisation du logiciel menant à l'ajout d'un nouvel utilisateur dans la base de données.

2.3.2. Diagramme de séquence

Suite aux cas d'utilisation et surtout grâce au scénario principal, il va être assez simple de produire des diagrammes de séquence. Ceux-ci vont représenter les interactions entre les acteurs et le système selon un ordre chronologique.

A = RecValue, TDate, PayeeTextField.getText(), DescriptionTextField.getText(), CategoryComboBox.getSelectedltem(), Amount, AccountsComboBoc.getSelectedlTem()

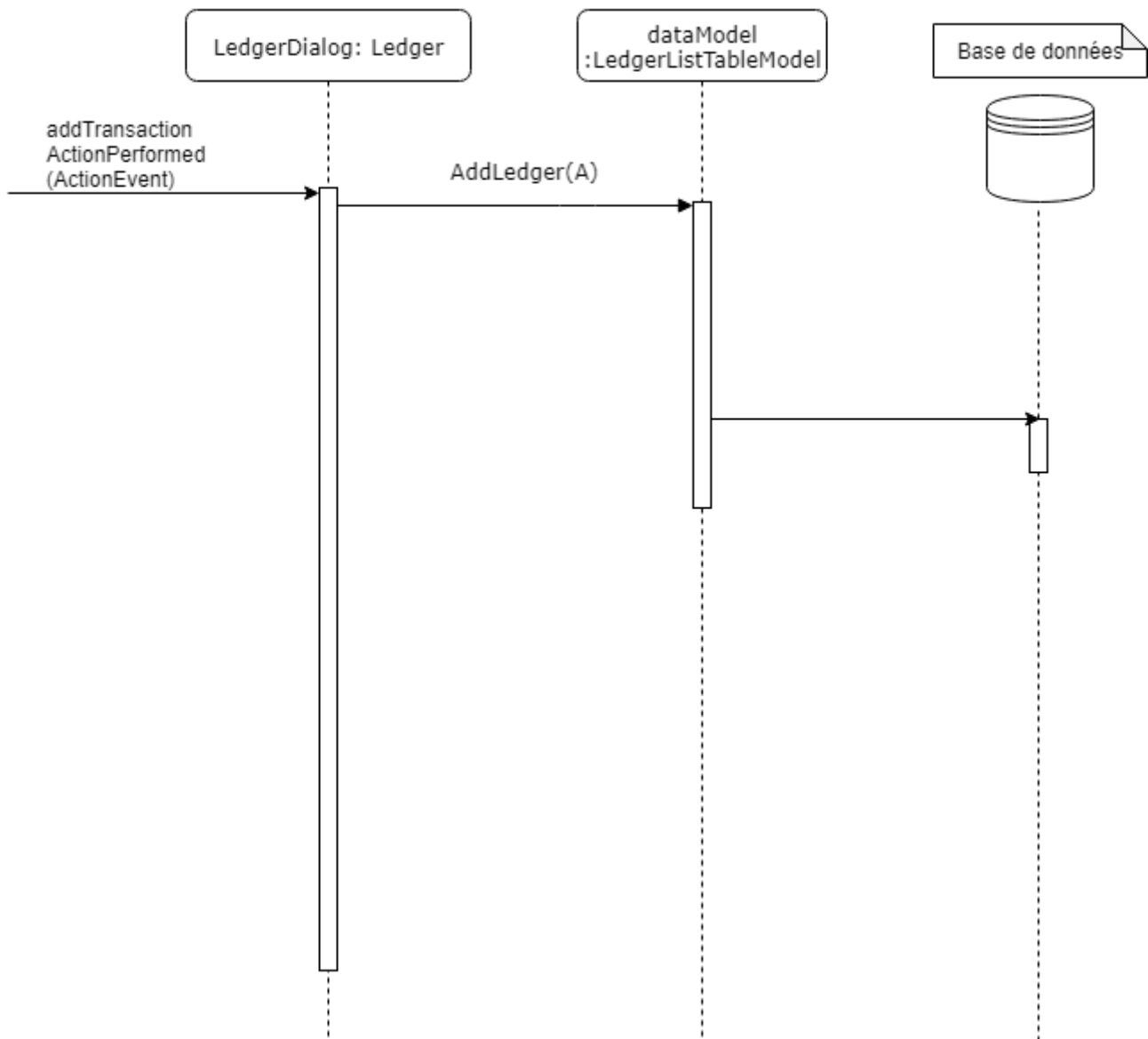


Figure 1. Diagramme de séquence de l'ajout d'un ajout dans la base

Ici, nous pouvons voir comment fonctionne l'ajout d'un champ dans la base (ici le champ Ledger). L'utilisateur appui sur le bouton d'ajout correspondant à la classe en question. Celle-ci va communiquer avec la classe chargée de la communication de la base de données qui va enfin insérer le nouveau champ dans la base.

2.4. Test de boîte noire

Maintenant que toutes les modifications, on été définies. Que nous connaissons les fonctionnalités qui en découlent. Et que nous connaissons les modules/classes concernés par les modifications. À débuté la phase de conception des tests.

Pour ce faire, nous avons mis en place la technique des "tests de boîte noire". Les tests en « boîte noire » consistent à examiner uniquement les fonctionnalités d'une application, c'est-à-dire si elle fait ce qu'elle est censée faire, peu importe comment elle le fait. Sa structure et son fonctionnement interne ne sont pas étudiés. Le testeur doit donc savoir quel est le rôle du système et de ses

fonctionnalités, mais ignore ses mécanismes internes. Il a un profil uniquement « utilisateur ».

Ainsi, cette méthode sert à vérifier, après la finalisation d'un projet, si un logiciel ou une application fonctionne bien et sert efficacement ses utilisateurs. Ce qui est cohérent par le fait que le logiciel initial est déjà fonctionnel, et que de nombreuses fonctions ne seront pas modifiées lors du développement.

Nous avons donc prévus de tester un programme en vérifiant que les sorties obtenues sont bien celles prévues pour des entrées données. Pour cela, nous avons isolé les variables critiques, celles qui pour certains paramètres provoquent des erreurs. Ces valeurs, provoquant des erreurs, sont les "valeurs frontalières".

2.4.1. Valeurs frontalières

Ces valeurs frontalières sont les valeurs qui sont aux limites de la création d'erreurs dans le logiciel. Par exemple, si une variable ne peut dépasser 50 caractères, les valeurs frontalières sont : 50 car jusque-là, il n'est pas censé avoir d'erreur. Et 51 car c'est à partir de là que le logiciel va produire une erreur (cet exemple est simplifié, il y a en général plus de valeurs frontalière par variable).


Description	Validité
"Name" possède au moins 1 caractère alphanumérique	valide
"Name" possède aucun caractère alphanumérique	invalide
"Name" est d'une longueur plus petite ou égale à 50	valide
"Name" est d'une longueur plus longue que 50	invalide
"Name" est unique	valide
"Name" n'est pas unique	invalide
"Description" est d'une longueur plus petite ou égale à 250	valide
"Description" est d'une longueur plus longue que 250	invalide
"Budget" est une valeur numérique (float)	valide
"Budget" n'est pas une valeur numérique (float) 	invalide
"Budget" est contraint à un nombre de 20 chiffres	valide
"Budget" est un nombre de plus de 20 chiffres	invalide

Figure 2. Ensemble des valeurs frontalières de la classe "Account"

Voici les valeurs frontalières pour une des classes du programme. Elles sont toutes aux bornes, inférieurs et supérieurs des erreurs.

2.4.2. Classes d'équivalence

Une fois toutes les valeurs frontalières définies. Nous avons produit un tableau de classes

d'équivalence. Le but d'une classe d'équivalence est de prendre une valeur frontalière défini plus tôt et de lui attribuer une validité ou non. C'est-à-dire, pour telle valeur de la variable est ce que la fonction va réussir à effectuer ce pourquoi elle programmée.

Le tableau de classes d'équivalence va donc permettre de définir une validité (ou non) pour chacune des valeurs frontalières.

Variables	Description	Identification des Classes d'équivalence	Valeurs frontalières	Validité	Justification
Name	Prénom et Nom du détenteur du compte	EC1	Name possède au moins 1 caractère alphanumérique	valide	Un Prénom et Nom valides peuvent contenir entre 0 et 50 caractères alphanumériques
		EC2	Name possède aucun caractère alphanumérique	invalide	Un Prénom et Nom valides doivent contenir au moins 1 caractère alphanumérique

Figure 3. Extrait d'une classe d'équivalence

Nous pouvons voir, de gauche à droite :

- Le nom de la variable concerné.
- Une description de cette variable.
- L'identificateur que l'on donne à cette classe d'équivalence.
- La valeur frontalière concernée.
- La validité du programme avec cette valeur.
- Une description permettant de comprendre la validité ou non de cette classe d'équivalence.

2.4.3. Cas de tests

Grâce aux classes d'équivalence, des tableaux de tests seront établis. Ces tableaux permettront d'établir les tests qui seront implémentés en reprenant les valeurs frontalières des classes d'équivalence. Voici un extrait d'un tableau de test :

Identification des tests	Pré Conditions	Post Conditions	Entrées nécessaires	Classes d'équivalence Valides	Classes d'équivalence Invalides	Sortie Obtenue	Explications
testAddAccountWithToolLongName	Aucunes	Message d'erreur, il n'y a pas de nouveau compte inséré dans la base	Le champ "Name" rempli avec plus de 50 caractères	EC1 EC3 EC5 EC8 / EC9	EC4	Insertion failed	La base de données tolère des noms uniquement inférieurs à 50 caractères

Figure 4. Extrait d'un tableau de cas de test

Nous pouvons voir, de gauche à droite :

- Le nom du test.

- Les préconditions pour effectuer le test.
- Les post-conditions permettant de définir si le test s'est bien déroulé.
- Les entrées des variables (ici, on veut qu'à l'entrée de la fonction, la variable "Name" ai plus de 50 caractères).
- L'identificateur de la classe d'équivalence définit plutôt. On met ici toutes les classes d'équivalence valides, c'est-à-dire celle qui ne produisent pas d'erreur. Dans ce cas, nous avons au moins 1 caractère alphanumérique (EC1), une longueur plus petite ou égale à 50 (EC2), etc.
- L'identificateur de la classe d'équivalence définit plutôt. On met ici toutes les classes d'équivalence invalides, c'est-à-dire celle qui produisent des erreur (cette colonne peut être vide si le test a pour but de ne pas produire d'erreur). Dans notre cas, nous avons une longueur plus longue que 50 (EC4), ce qui va produire une erreur.
- La sortie système obtenu. Ici, c'est ce que le logiciel va retourner à l'utilisateur dans le terminal.
- Une description permettant d'expliquer le déroulement du test.

Le but de ces cas de tests est de couvrir toutes les classes d'équivalence invalides pour être sûr de tester tous les cas d'erreur possible (ou au moins ceux définît par l'équipe) tout en évitant la redondance. Une fois, toutes les classes d'équivalence invalide traitées, il ne reste plus qu'à les implémenter.

2.5. Phase de Développement

Nous allons ici nous attarder sur les principaux outils de développement utilises. Nous verrons ensuite les résultats de la phase de programmation. La programmation en elle-même n'étant pas très intéressante (programmation classique en java EE).



En ce qui concerne la chronologie du projet, la phase de développement n'a pas débuté après la phase de conception présentée plus tôt. Elle était en réalité faite en parallèle. Cependant, elle comprenait des parties de développement mineurs qui n'ont pas nécessité de conception. Comme de la correction ou de l'optimisation de code. C'est pendant ces phases de développement mineur que nous avons travaillé sur la conception. Je vais donc ici m'attarder sur les points centraux du développement.

2.5.1. Trac

Avant de se lancer dans la programmation, nous avons mis en place un service Trac.

Trac est un wiki amélioré et un système de suivi des problèmes pour les projets de développement logiciel. Il fournit une interface à Git (ou d'autres systèmes de contrôle de version), un wiki intégré et des fonctions de rapport pratiques. Trac permet de baliser le wiki dans les descriptions de problème et de valider des messages, en créant des liens et des références transparentes entre les bogues, les tâches, les ensembles de modifications, les fichiers et les pages wiki. Une chronologie affiche dans l'ordre tous les événements de projet actuels et passés, ce qui facilite l'acquisition d'une vue d'ensemble du projet et le suivi de ses progrès. La feuille de route indique la voie à suivre, en énumérant les jalons à venir.

Trac permet aussi de mettre au point des "tickets" : ceux sont des représentations des tâches à effectuer.

Ticket #26 (new bug-report)

There is a Big Bug in the figgle flop. Opened 17 hours ago

Reported by: mark

Owned by:

Priority: high

Milestone:

Component: libs

Version: 1.0

Keywords:

Cc:

Parent ID:

Description

Although we are running version 1.0 of the figgle flop, this bug appears in all versions we know of. Reply

Child Tickets Create

Ticket	Type	Priority	Owner	Summary	Milestone
#27	bug-fix	highest		Bugfix for the figgle flop issue.	FIGFLOP-BUGFIX
#28	bug-fix	highest	mark	Fix the figgle flop bug on (unreleased) version of product.	FIGFLOP-2.0
#29	bug-fix	low		Fix the figgle flop bug on 1.x version of product.	FIGFLOP-1.1

Figure 5. Exemple d'un ticket Trac

On peut ainsi notamment y voir : la ou les personnes assignées à cette tâche, la priorité de la tâche ou encore le jalon auquel le ticket appartient.

Dans ce projet, Trac va servir de plateforme principale pour tout ce qui tourne autour de la programmation :

- Elle va regrouper une partie de la documentation. C'est par exemple tout ce qui touche à la conception ou au fonctionnement du logiciel.
- Elle permet de renvoyer vers le GIT du projet, mais aussi vers les rapport PMD (que nous allons voir juste après).
- Elle regroupe les tickets en cours et ceux réalisés

Une fois cet outil mit en place, nous nous sommes tournés vers la mise en place de QALab.

2.5.2. QALab

QALab est un plugin de Maven, il permet de générer de la documentation.

Nous avons avant tout utilisé deux outils de QALab :

Checkstyle : outil qui permet une validation du style de code et une vérification de la conception.

Down by 7 file(s), Down by -12 Violations

File Name	Type	Previous Run	Previous Value	Current Run	Current Value	Diff
ca/etsmtl/log240/financej/Account.java	checkstyle	2019-01-14	507	2019-01-25	506	-1
ca/etsmtl/log240/financej/App.java	checkstyle	2019-01-14	9	2019-01-25	3	-6
ca/etsmtl/log240/financej/Category.java	checkstyle	2019-01-14	178	2019-01-25	177	-1
ca/etsmtl/log240/financej/DBUtils.java	checkstyle	2019-01-14	45	2019-01-25	44	-1
ca/etsmtl/log240/financej/FinanceJ.java	checkstyle	2019-01-14	236	2019-01-25	235	-1
ca/etsmtl/log240/financej/Ledger.java	checkstyle	2019-01-14	349	2019-01-25	348	-1
ca/etsmtl/log240/financej/Reports.java	checkstyle	2019-01-14	200	2019-01-25	199	-1

Figure 6. Erreurs trouvées par Checkstyle

Ceci est le résultat d'une session de correctifs d'erreurs décelées par Checkstyle. Chacune de ces classes présentait un certain nombre d'erreurs, elles sont présentes dans la colonne Previous Value. On peut ensuite voir que pour la version actuelle du projet, ici celle du 25 janvier 2019, le nombre d'erreurs à baisser. Le nouveau nombre d'erreurs est situé dans la classe Current Value. Nous avons donc sept classes Java qui ont perdu au moins une erreur chacune dans leur code.

PMD : outil qui permet une vérification du code (bugs possibles, code mort, code sous-optimal, etc.).

Up by 7 file(s), Up by 1622 Violations

File Name	Type	Previous Run	Previous Value	Current Run	Current Value	Diff
ca/etsmtl/log240/financej/Account.java	pmd	-	0	2019-01-30	17	17
	checkstyle	-	0	2019-01-30	506	506
ca/etsmtl/log240/financej/App.java	pmd	-	0	2019-01-30	1	1
	checkstyle	2019-01-25	3	2019-01-30	4	1
ca/etsmtl/log240/financej/Category.java	pmd	-	0	2019-01-30	18	18
	checkstyle	-	0	2019-01-30	177	177
ca/etsmtl/log240/financej/DBUtils.java	checkstyle	-	0	2019-01-30	44	44
ca/etsmtl/log240/financej/FinanceJ.java	pmd	-	0	2019-01-30	26	26
	checkstyle	-	0	2019-01-30	235	235
ca/etsmtl/log240/financej/Ledger.java	pmd	-	0	2019-01-30	29	29
	checkstyle	-	0	2019-01-30	348	348
ca/etsmtl/log240/financej/Reports.java	pmd	-	0	2019-01-30	21	21
	checkstyle	-	0	2019-01-30	199	199

Down by 1 file(s), Down by -12 Violations

File Name	Type	Previous Run	Previous Value	Current Run	Current Value	Diff
ca/etsmtl/log240/financej/DBUtils.java	pmd	2019-01-25	19	2019-01-30	7	-12

Copyright © 2019 ÉTS - LOG240 - Team 25. All rights reserved.

Figure 7. Erreurs trouvées par PMD

Même explication ici, dans la partie Up by, nous pouvons voir les classes qui comportent des erreurs dans leur code, mais qui n'ont pas été modifiées lors de la précédente mise à jour du projet. Dans la partie Down by, nous pouvons voir que les correctifs de la dernière mise à jour se sont concentrés sur des erreurs repérées par PMD dans la classe DBUtils.java. En effet, lors de la précédente version du projet, le nombre d'erreurs (Previous Run) était de 19. Après les correctifs apportés par la dernière version du projet, le nombre d'erreurs est passé à 7 (Current Value).

Pour chacun de ces deux outils, QALab va générer des diagrammes permettant de voir l'état actuel du projet. Que ce soit le nombre d'erreurs actuel, mais aussi le nombre d'erreurs dans les versions précédentes du projet.

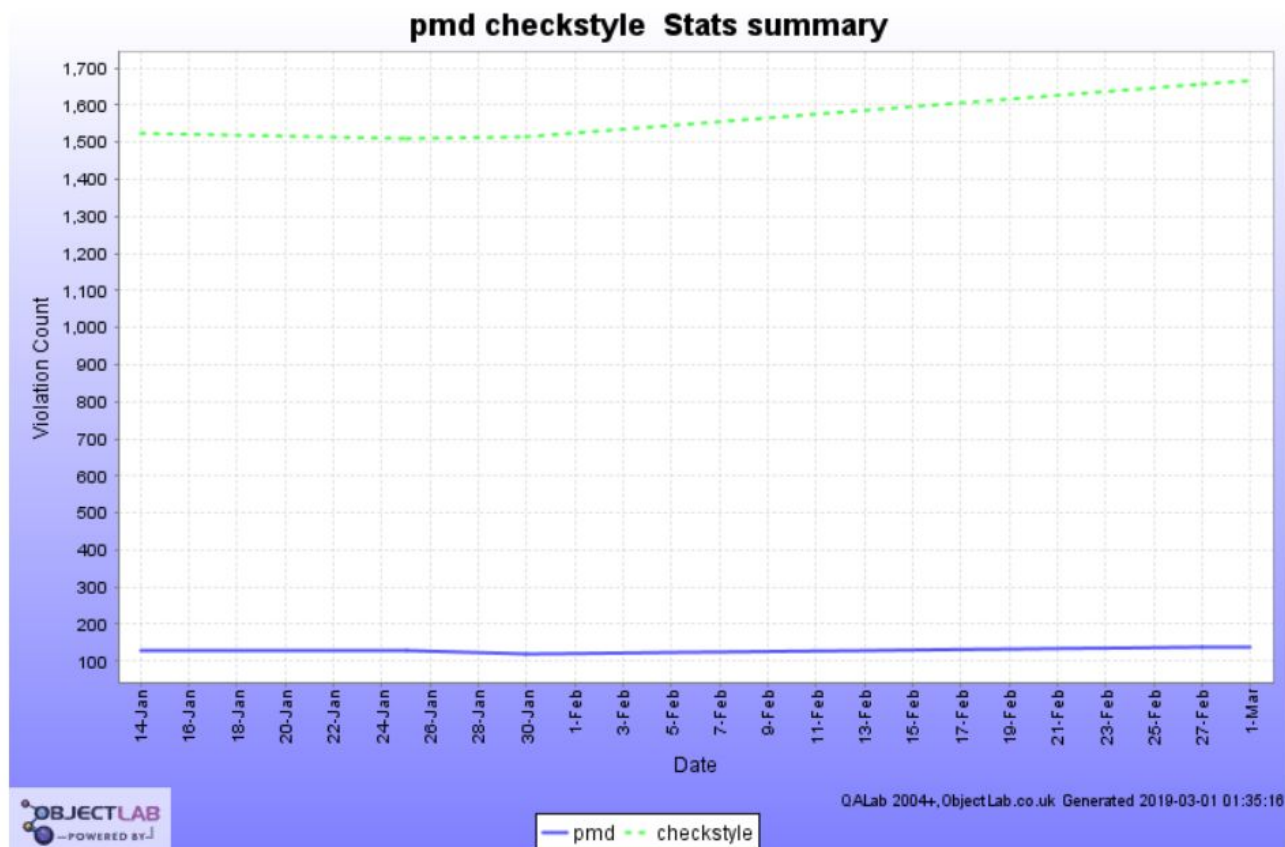


Figure 8. Rapport de QALab

Ici, nous pouvons voir les différentes sessions de corrections d'erreurs. Tout d'abord, le code possédait 1524 erreurs détectées par *Checkstyle*, ainsi que 131 erreurs détectées par *PMD*. Après la première session de correctifs que nous avons apportés, les erreurs de type *CheckStyle* sont passé de 1524 à 1512. Nous avons ensuite corrigé des erreurs soulevées par *PMD*, elles sont ainsi passées de 131 à 119, on peut cependant noter qu'une nouvelle erreur a été détectée par *Checkstyle*.

2.5.3. Tests

L'application proposée aux utilisateurs est une interface. Ce qui veut dire que toutes les opérations passent par elle. Nous avons donc mis en place des tests utilisant celle-ci.

Le test va donc remplir les champs de l'application et appuyer sur les boutons pour soumettre les requêtes.

```

public void testAddAccountWithTooLongName() throws Exception {

    WindowInterceptor.init(accountsButton.triggerClick())
        .process(new WindowHandler() {
            public Trigger process(Window window) {

                accountsTable = window.getTable();
                int initialRowCount = accountsTable.getRowCount();

                window.getTextBox( name: "NAME_TEXT_FIELD").setText("tooManyCharacterstooManyCharacterstooManyCharacters");
                window.getTextBox( name: "DESCRIPTION_TEXT_FIELD").setText("Try to add a too big name");

                //window.getButton("Add AccountDialog").click();

                WindowInterceptor.init(window.getButton( name: "Add AccountDialog").triggerClick())
                    .process(new WindowHandler() {
                        public Trigger process(Window window) {
                            System.out.println("On est bien dans le interceptor");
                            return window.getButton( name: "OK").triggerClick();
                        }
                    });

                assertEquals(accountsTable.getRowCount(), initialRowCount);

                return window.getButton( name: "Close").triggerClick();
            }
        }).run();
}

```

Figure 9. Test d'ajout dans la base d'un champ trop long

Ici, nous effectuons un test qui a pour but de générer une erreur. En effet, on tente d'ajouter un utilisateur avec un nom trop grand. On va donc vérifier qu'après avoir essayé d'ajouter cet utilisateur, une erreur est bien générée. Et que l'utilisateur n'a donc pas été ajouté dans la base.



"accountsButton" est en rouge, car dans la version final du projet, il n'existe plus. En effet, ces tests fonctionnent avant la réingénierie du projet qui a déplacé les interactions avec la base de données dans une classe spécifique (DAO).

2.6. Résultat

Maintenant que toutes les modifications ont été implémentées. Le logiciel possède une interface de connexion :

Figure 10. Fenêtre de connexion

On peut ici y renseigner son identifiant et son mot de passe. Le logiciel va ensuite limiter les droits en fonctions du poste occupé par la personne qui se connecte. Premier exemple, ici un utilisateur "employé" :

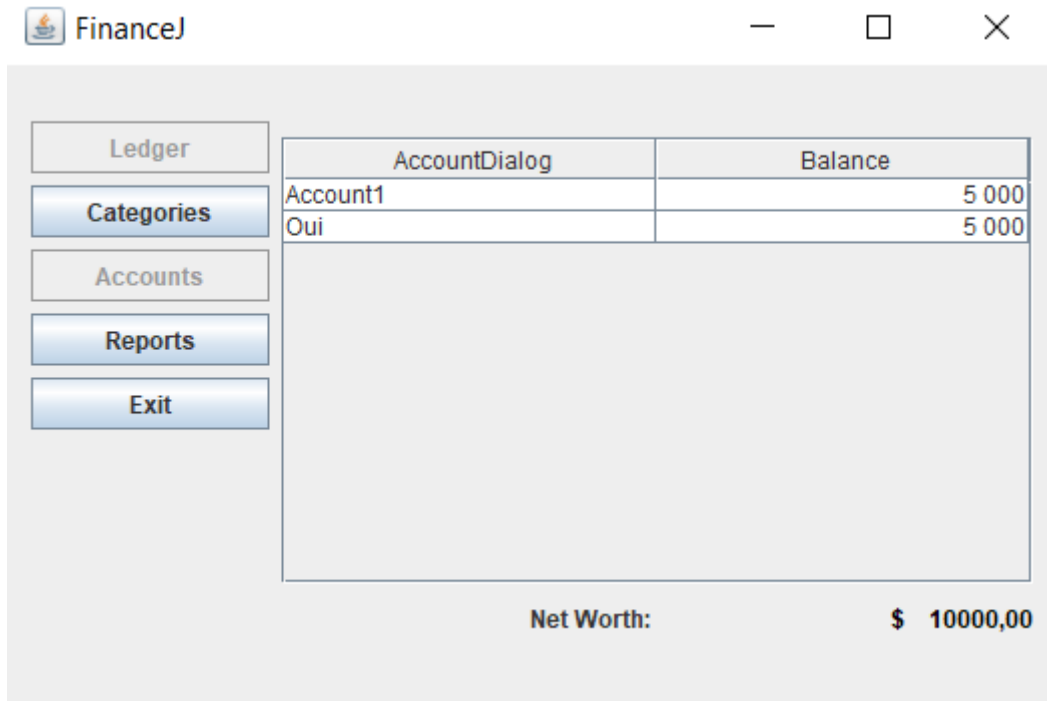


Figure 11. L'application d'un employé

Il n'a pas accès à toutes les fonctionnalités logiciel (ici Account et Ledger). Maintenant voyons la fenêtre d'un utilisateur ayant tout les accès :

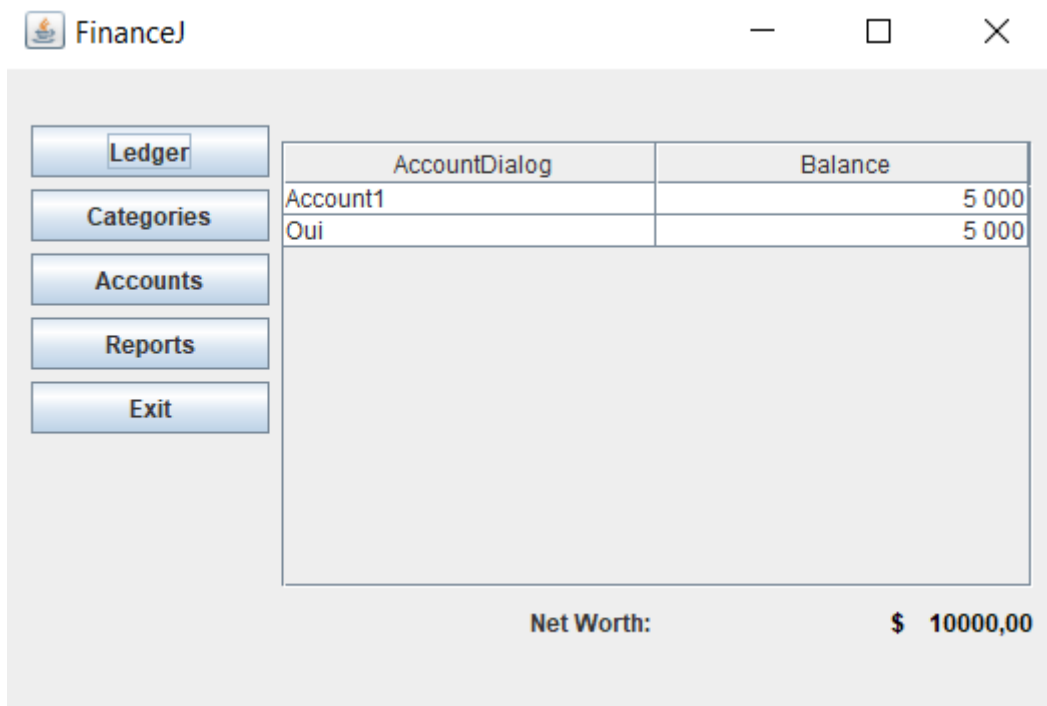


Figure 12. L'application d'un "chef"

Ici, aucune des fonctionnalités est indisponible.

2.7. Conclusion

Si j'ai choisi de vous présenter ce projet, ce n'est pas pour son côté programmation comme vous avez pu le constater. Il fait en effet suite au cours du semestre 4 concernant le développement en Java EE. C'est une partie de ce qui m'a plus, car ce projet m'a permis de mieux comprendre est

d'approfondir ce point que j'avais assez mal assimilé lors de mon DUT.

Si j'ai choisi de vous parler de ce projet, c'est pour tout ce qui gravite autour. J'ai appris plusieurs nouvelles méthodes qui servent à la conception. Que ce soient les cas d'utilisation, la définition des valeurs frontalières et la mise en place des classes d'équivalence.

Mais aussi pour les nouveaux outils de maintenance que je trouve très pratique. QALab dans un premier temps, qui relève énormément d'erreurs et de mauvaise pratique pouvant mener à de futures erreurs. Trac m'a aussi beaucoup plus, la création de tickets et pratique bien qu'un peu superficiel, cette tâche pouvant être réalisé sur n'importe quel document. Cependant, son wiki intégré est assez simple d'utilisation et à mettre en place.

Les tests par utilisation de l'interface graphique sont aussi une découverte pour moi. Bien que je trouve ceux-ci un peu limités car essayant de reproduire le comportement humain, qui est assez imprévisible. Je les trouve assez intuitifs.

3. Enrichissement culturel

3.1. Vie Universitaire

Au niveau des cours, il faut savoir que les Québécois peuvent choisir les cours qu'ils veulent suivre durant leur session parmi une liste de leur diplôme. En tant qu'étudiant étranger, j'ai donc pu piocher parmi ce qui m'intéressait. Les cours sont composés à moitié d'heures magistrales et à moitié d'heure de travaux pratiques, ce qui se rapproche beaucoup de l'IUT ici en France. Il y a des examens à la mi-session et en fin de session ainsi que des travaux pratiques ou des devoirs en fonction des cours. Le système de note est assez complexe (comme vu au chapitre 1).

Les relations avec les professeurs sont plus souples qu'en France. Ils sont ouverts à toutes questions et disponible pour donner de l'aide. Certains se font même appeler par leur prénom. Les professeurs se sentent proches des élèves et il n'y a pas de cadre très formel. Tout le monde se tutoie, mais il peut parfois être choquant pour nous étudiant français de voir un élève québécois mécontent face à un professeur.

Les cours dans leur globalité sont assez trompeurs pour les étrangers, car ils sont souvent très complets. On commence les premières semaines par des choses simples, mais cela devient très vite pour complexe et il ne faut pas se laisser aller dès le début, car c'est difficile de se rattraper par la suite.

L'école de technologie supérieure est un très grand institut qui accueille énormément d'étudiants étrangers à chaque semestre. Le système administratif est assez bien organisé, je n'ai pas vraiment eu de problème.

3.2. Vie quotidienne

Le Québec est connu pour son hiver plutôt froid. Il faut s'y préparer, car il dure une grande partie de l'année (novembre à avril environ). La météo annonce toujours deux températures : la température réelle, et la température ressentie, qui est plus froide, à cause du facteur vent. Ainsi, -26° peuvent être ressentis comme -40°. Mais une fois qu'on est équipé, la température n'est plus un problème. Et beaucoup d'activités sont proposées l'hiver, que ce soit par l'université ou par la ville.

Au niveau du rythme de vie, il ressemble dans l'ensemble à celui des Français. La différence flagrante est au niveau du repas du soir. En effet, au Québec, le souper se prend tout de suite après la fin du travail. Ainsi, les Québécois mangent entre 17 h et 18 h généralement.

Les transports en commun sont très répandus et fréquents. La Société de Transport de Montréal (STM) opère 165 lignes de bus et quatre lignes de métro sept jours sur sept. Il est possible de prendre des bus vingt-quatre heures sur vingt-quatre. En revanche, les métros s'arrêtent aux alentours d'une heure du matin. Les magasins sont ouverts tous les jours, en général jusqu'à dix-neuf heures, jusqu'à seize heures, le week-end.

Vous trouverez des "dépanneurs", petites épiceries, un peu partout ouvertes vingt-quatre heures sur vingt-quatre, mais pratiquant des prix plus élevés. Le centre-ville regorge de grands centres commerciaux dont une partie est souterraine afin de les relier entre eux et au métro.

Au niveau de la nourriture, on est en Amérique donc des concessions sont à faire sur certains points. Mais on trouve vite son compte. Il n'y a pas de grandes spécialités au-delà de la poutine et du sirop d'érable ou de plats à base de ce produit local, mais en général, on y mange bien. À savoir que les pharmacies ne vendent pas que des médicaments, mais aussi des cosmétiques et de la nourriture.

3.3. Bilan personnel

De part la francophonie du Québec je n'ai que peu mis à l'épreuve mon anglais. Cependant, dans ma collocation, il y a toujours eu des colocataires d'origine allemande. Et nous nous sommes attachés à parler anglais entre nous.

De plus, pendant les quelques vacances scolaire, j'ai pu voyager à New-York ainsi qu'à Toronto. Ces deux séjours de courtes durées, au-delà de leurs aspects touristiques, m'ont bien servi pour tester mes capacités à parler anglais toute la journée.

Je tire un bilan très positif de mon année à l'étranger. En effet, il n'est pas très facile au début de se retrouver loin de sa famille ainsi que de ses amis. Néanmoins, je ne tire que du positif de cette expérience, en effet, je suis devenu bien plus mature qu'auparavant et bien plus autonome. D'une manière générale, je dirais ce séjour a fait naître en moi l'envie de voyager et de découvrir le monde. Je n'étais encore jamais parti aussi loin et aussi longtemps de la France. J'ai appris à me débrouiller par moi-même, à être un peu plus curieux de ce qui m'entoure et à m'ouvrir aux autres. Ma vision de la vie a sans doute évolué au court des mois. Je ne rentre pas en France avec les mêmes envies et les mêmes attentes, mais c'est une bonne chose.

Conclusion

Malgré quelques soucis, notamment le fait que je ne peux pas valider le DUETI, si cette année était à refaire,. Peu importe les erreurs, cette expérience a été la plus incroyable de ma vie et je conseillerai à n'importe qui de se lancer si cela était à faire. Je suis parti relativement seul, et je rentre avec plein de nouvelles connaissances, de souvenirs et de vécu, avec une perception un peu change de ma vie. Je regrette cependant d'être resté majoritairement avec des Français pendant un moment avant de m'ouvrir. Si pareille occasion venait à se présenter à moi dans le futur, j'essayerais de rencontrer plus de gens et de plus visiter, sans doute.

Je prévois maintenant de reprendre mon cursus en France. Je serai en Licence 3 l'année prochaine, et je ne sais pas encore si je compte poursuivre sur un master ou en rester là.

Annexes

Annexe 1

CU1 - Créer un compte

Préconditions

- L'utilisateur est connecté.
- L'utilisateur est dans l'onglet "Account".

Scénario principal

1. L'utilisateur entre le nom dans le champ "Name".
2. L'utilisateur remplit ou non le champs "Description", ce champ est facultatif.
3. L'utilisateur appuie sur le bouton "Add Account".
4. Le système enregistre le compte avec les données entrées.
5. Le système affiche le compte créé dans la liste de comptes.

Scénarios alternatifs

- L'utilisateur n'a pas inscrit de nom dans le champ "Name".
 - Le système affiche un message d'erreur indiquant que le nom est obligatoire.
 - Retour à l'étape 1.
- L'utilisateur inscrit un nom qui se trouve déjà dans la liste.
 - Le système affiche un message d'erreur indiquant que le nom doit être unique.
 - Retour à l'étape 1.