

Université Paris 8 – Vincennes – Saint-Denis

2 Rue de la Liberté

93526 Saint-Denis CEDEX

ZURAWSKA Alexis

12310497

Rapport de stage



Site web : <https://www.univ-paris8.fr/>

Accueil Handicap

Elka PARVANOVA

0149407147

eparvanova@univ-paris8.fr

Vincent BOYER

0149406403

boyer@ai.univ-paris8.fr

Commencement du développement d'un logiciel de gestion des informations des étudiants handicapés de l'université
Paris 8

1^{er} juin 2016-31 août 2016

Table des matières

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Résumé | 1 |
| Summary..... | 2 |
| Remerciements | 3 |
| Introduction | 4 |
| I Paris 8 : la grande université de la mixité à la grande histoire malgré son jeune âge et au panel de formations diverses et variées | 6 |
| A Un panel de diverses formations proposées sur plusieurs campus..... | 6 |
| 1) Brève histoire et chiffres de l'université Paris 8 | 6 |
| 2) Présentation des différentes formations et des différents campus..... | 8 |
| B Une ouverture sur le monde permettant une mixité sans égale..... | 9 |
| 1) Un partenariat avec des universités mondialement reconnues..... | 10 |
| 2) Des étudiants de tous les horizons étudient au sein de l'université | 11 |
| C Une politique sur le handicap en avance sur d'autres universités | 12 |
| 1) Présentation de la loi du 11 février 2005, du service et de ses missions | 12 |
| 2) Position occupée au sein du service | 14 |
| II Un projet de développement très utile et pratique pour l'accueil-handicap..... | 15 |
| A Présentation de l'application et de son fonctionnement | 15 |
| 1) Présentation de l'application..... | 15 |
| 2) Présentation du fonctionnement de l'application..... | 16 |
| B Des contraintes précises à respecter rendant le projet plus difficile que prévu | 19 |
| 1) Développement initialement prévu | 19 |
| 2) Des difficultés causées par les contraintes empêchant le développement initial..... | 21 |
| C Le langage java : un choix stratégique par rapport à d'autres langages | 22 |
| 1) Un langage multiplateforme très utilisé sur le marché du travail..... | 22 |
| 2) Un paradigme de programmation rendu relativement simple par le langage : la POO | 25 |
| III Un stage qui se différencie des précédents | 28 |
| A Un mode de travail très différent... | 28 |
| 1) Un stage réalisé à partir de rien, grâce aux acquis des stages précédents | 28 |
| 2) La découverte du télétravail..... | 30 |
| B ...Nécessitant une grande autonomie | 31 |
| 1) Un stage réalisé grâce à des pistes..... | 31 |
| 2) ...Tel un projet d'université..... | 32 |
| C L'après stage : un code qui se veut le plus clair et lisible possible..... | 33 |
| 1) Deux documentations techniques accompagnent le projet..... | 33 |
| 2) Un code très, voire trop, concis..... | 34 |
| Conclusion..... | 36 |
| Annexe A : Organigramme du service | I |
| Annexe B : Glossaire des termes techniques..... | II |
| Annexe C : Classe Fenetre.java au début du développement | V |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Annexe D : Classe Fenetre.java modifiée par la suite + quelques classes pour montrer l'évolution du code : | VII |
| Annexe E : Documentation technique du projet | XI |
| Annexe F : Morceaux de la javadoc du projet | XIX |
| Annexe G : Journal de bord du stage | XXI |
| Annexe H : Bibliographie/sitographie/filmographie..... | XXIV |
| Annexe I : Bugs non résolus à ce jour..... | XXV |
| Annexe J : Quelques captures d'écrans des écrans de l'application | XXVI |
| Annexe K : Captures d'écrans de deux fichiers de données (ici, identité et projets) utilisés pour le développement de l'application | XXXIV |
| Annexe L : captures d'écran des fichiers departements et statut pour remplir des listes déroulantes de certains écrans de l'application | XXXVII |

Résumé

Etudiant en informatique, je devais, dans le cadre de ma première année de Master au sein de l’UFR¹ MITSIC² de l’université Paris 8 Vincennes – Saint-Denis, effectuer un stage d’une durée de trois mois minimum. Ce stage devait avoir pour but de me faire approfondir les concepts et méthodes étudiés en cours en plus d’en apprendre de nouveaux. Ce stage devait faire l’objet d’un rapport écrit ainsi que d’une soutenance orale devant un jury.

Après un début de stage difficile au Ministère de la Justice, entre début avril et mi-mai 2016, ayant eu pour conséquence une rupture de stage, l’accueil handicap de l’université m’a proposé, avec l’accord de mon responsable pédagogique, de me recruter pour un stage de trois mois entre le 1^{er} juin et le 31 août 2016 afin que je puisse soutenir mon travail et ne pas être obligé de redoubler mon année.

Je devais, dans le cadre de ce stage, commencer à développer une application de gestion des informations des étudiants handicapés de l’université, permettant de faciliter le travail du personnel du service quant à la création de nouveaux dossiers d’étudiants handicapés ou à la mise à jour des existants.

Pour se faire, j’ai utilisé mon propre ordinateur que j’ai équipé de l’IDE³ Eclipse pour le développement en langage java, ainsi que du pack Office pour la création de fichiers d’informations fictifs d’étudiants handicapés au format csv avec Microsoft Excel et pour la rédaction de la documentation technique avec Microsoft Word. Par ailleurs, j’ai utilisé l’éditeur de texte Notepad++ pour écrire les fichiers txt servant, entre autres, à remplir les listes déroulantes de certains écrans de l’application. Chacun des fichiers csv représente les informations que devait contenir un écran précis de l’application en consultation.

Actuellement, seule la consultation est possible et certains écrans sont en cours de développement ou à développer.

¹ Unité de Formation et de Recherche

² Mathématiques, Informatique, Technologies et Sciences de l’Information et de la Communication

³ cf : glossaire

Summary

Student in IT⁴, I had to, in the context of my First year of Master on Paris 8 Vincennes – Saint-Denis university MITSIC⁵ Formation and Search United, effectuate a training which duration was minimum three months. This training had to goal to me to deepen concepts and methods studied in course and learn others in enterprise context. This had to subject to a written report and a defense in front of jury.

After a difficult starting of training in Paris Ministry of Justice between the beginning of 2016 Appril and the First half of 2016 May, have resulted a breaking of my training, handicap home of the university propose to me to recruit for a three months training between June 1st 2016 and August, 31st, with accord of Vincent BOYER, head teacher of my formation, so that I don't redouble my year because of the short duration of my previous training.

During this training, I had to start to developpe an application of management of disabled students informations to facilitate work of staff to create new student folder or update existants.

To realize this, I used my own computer with Eclipse IDE for the java developpment, Microsoft Excel for the creation of csv files containing students information (one file for each screen of application for the consultation) and Microsoft Word for the redaction of technical documentation of the project. Otherwise, I used Notepad++ text editor to create txt files for fill combo box.

Actually, only consultation is possible and some screen are being developpement or to developp.

⁴ IT = Information Technology = Informatique

⁵ cf : note 2

Remerciements

Je tiens, tout d'abord, à infiniment remercier Elka PARVANOVA qui a réussi, avec succès, à me délivrer d'un précédent stage qui a mal fini en plus de me permettre d'effectuer mon stage au sein de son service, afin que je ne sois pas pénalisé et obligé de redoubler ma première année de Master informatique à cause de la trop courte durée de mon premier stage par rapport à celle requise.

Un grand merci à Danielle TARTAKOWSKY, présidente de l'université Paris 8, ainsi qu'à Arab ALI CHERIF, responsable de l'UFR MITSIC, pour leur collaboration à la finalisation du stage.

Enfin, un merci tout particulier à Vincent BOYER, responsable pédagogique du M1 informatique de Paris 8, pour avoir collaboré avec l'ensemble de l'équipe du service handicap, que je remercie beaucoup également, pour que je commence mon nouveau stage, en plus de me donner de bonnes pistes pour le commencement du développement du logiciel sur lequel j'ai dû travailler.

Introduction

Dans le cadre de ma première année de Master en informatique, effectuée au sein de l’UFR MITSIC de l’université Paris VIII – Vincennes – Saint-Denis, je devais effectuer un stage obligatoire d’une durée minimum de 3 mois, afin d’approfondir les concepts et méthodes étudiés en cours et d’en apprendre de nouveaux dans le cadre professionnel. Les cours finissant à la fin du mois de mars 2016, nous pouvions commencer notre stage dès début avril et nous pouvions étendre ce stage jusqu’à fin août. Ayant effectué un stage d’une durée de deux mois et demi lors de ma seconde année de DUT informatique à l’IUT de Montreuil ainsi qu’un autre d’une durée de trois mois l’année dernière, dans le cadre de ma licence, je voulais, cette année, effectuer un stage d’au moins quatre mois. Après avoir postulé à plusieurs offres auprès de grands groupes tels que BNP Paribas, Société Générale, etc..., j’ai fait appel aux parents d’un ami afin de savoir s’ils avaient une opportunité, sur leurs lieux de travail respectifs, ou s’ils connaissaient quelqu’un qui serait intéresser pour me recruter. C’est ainsi que j’ai été contacté par un ami de ces amis. Après l’envoi d’un CV et d’une lettre de motivation à ce monsieur, qui travaillait au Ministère de la Justice, celui-ci m’a convoqué pour un entretien avec les ingénieurs informaticiens qu’il avait sous sa direction. L’entretien s’était très bien passé, et je devais, initialement, effectuer un stage de quatre mois entre le 4 avril et le 5 août 2016. Mais après un mois et demi de stage difficile, j’ai été obligé de rompre mon stage. C’est ainsi que, n’étant pas soutenable, compte tenu de la durée trop courte effectuée par rapport à celle requise, Vincent BOYER, responsable pédagogique du M1 informatique de l’université, avec l’accord de la cellule handicap, m’a proposé d’effectuer un nouveau stage d’une durée de trois mois entre le 1^{er} juin et le 31 août 2016.

L’accueil handicap de l’université gère, pour le moment, les dossiers des étudiants handicapés au format papier. Ces dossiers sont remplis lors de la première visite chez eux et sont mis à jour tous les ans. Les informations contenues dans ces dossiers sont ensuite synthétisées dans un très grand fichier Excel.

Or, cette solution se révèle difficile pour retrouver les informations d'un étudiant en particulier pour mettre son dossier à jour, sans parler des dossiers papiers qui sont parfois difficiles à retrouver malgré un classement le plus méticuleux possible. De fait, l'équipe avait besoin d'une solution informatisée qui permettrait, au moyen de la recherche par numéro d'étudiant, nom et/ou prénom, de trouver un unique étudiant et d'accéder à ses informations (identité, projets, aides accordées par la MDPH, etc...)⁶ pour éventuellement les mettre à jour, en plus de permettre la création de nouveaux dossiers.

C'est après vous avoir présenté la brève histoire et la constitution de l'université que je vous présenterais, de manière détaillée, l'application sur laquelle j'ai dû travailler. Pour finir, je montrerai les spécificités de ce stage par rapport aux deux autres précédents.

⁶ cf : documentation technique du projet, annexe E p XI-XVIII

I Paris 8 : la grande université de la mixité à la grande histoire malgré son jeune âge et au panel de formations diverses et variées

Il s'agira, dans cette première partie, de présenter l'université Paris 8 à travers son historique, son panel de formations sur ses différents campus et sa politique d'ouverture sur le monde et sur le handicap.

A Un panel de diverses formations proposées sur plusieurs campus

Il sera question, ici, de brièvement présenter l'histoire de l'université avant de présenter les différents campus ainsi que les formations qui y sont dispensés.

1) Brève histoire et chiffres de l'université Paris 8

Au mois de mai 1968, alors qu'une révolte antiautoritaire contre le pouvoir gaulliste sévit, une grande manifestation étudiante et ouvrière se déroule à la Sorbonne, dénonçant, entre autres, le caractère élitiste de l'accès aux études supérieures. C'est dans ce contexte que le général De Gaulle décide de leur offrir une faculté dotée d'une grande autonomie pédagogique et financière. Cette université voit le jour en janvier 1969 mais elle n'ouvre officiellement ses portes que le 3 mai de cette même année. Elle est connue, à l'époque, sous le nom de CUEV⁷. Ce centre Expérimental est alors situé dans le bois de Vincennes sur le sol parisien et devait, initialement, être une annexe de la Sorbonne. Toutefois, cette université se voulait indépendante et unique en son genre du fait qu'elle soit ouverte à tout le monde.

⁷ Centre Universitaire Expérimental de Vincennes

En 1971, elle change de nom et devient, désormais, l'université Paris 8 Vincennes. L'idée de cette université était de permettre, à l'époque, à toute personne, sans limite d'âge ni condition de diplôme, d'accéder aux études supérieures, notamment via le système des cours du soir. L'esprit de l'université de Vincennes faisait perdurer l'idée du mouvement de Mai 1968.

Initialement conçue pour accueillir 8 000 étudiants, le pôle universitaire s'est retrouvé à accueillir quatre fois ce nombre d'étudiants, du fait de son succès dont elle a été victime. D'autant que, dans les années 1975-76, il se déroulait des trafics de drogues et des dégénérescences des locaux, causées par les étudiants. Puis, en 1980, suite à la décision de Jacques Chirac, maire de Paris à l'époque, de récupérer les terrains occupés par l'université, celle-ci a dû déménager à Saint-Denis, d'où sa nouvelle appellation d'université Paris 8 – Vincennes à Saint-Denis. Aujourd'hui, aucune trace des anciens locaux de Vincennes ne subsiste.

En 2016, l'université compte 21 846 étudiants, parmi lesquels 14 042 sont inscrits en licence, 6 059 en master, 991 en doctorat, 747 dans des diplômes hors LMD⁸, etc...

L'université Paris 8, c'est aussi 271 partenariats avec des universités mondialement reconnues, 145 nationalités représentées parmi les étudiants, 899 enseignants, tous grades confondus et 868 personnels BIATOS⁹.

⁸ Cursus Licence-Master-Doctorat

⁹ Bibliothécaires, Ingénieurs, Administratifs, Techniciens, Ouvriers, de Services Sociaux et de Santé

2) Présentation des différentes formations et des différents campus

L'université Paris VIII – Vincennes – Saint-Denis offre un large panel de formations variées dans six domaines, au travers de ses 11 UFR se trouvant sur son campus principal de Saint-Denis, allant de la licence (dont des licences professionnelles) au doctorat :

- lettres et langues (UFR langues et cultures étrangères (LLCE¹⁰-LEA¹¹)) ;
- sciences humaines et sociales (UFR textes et société, UFR ERITES¹², UFR psychologie, UFR SEPF¹³, UFR SDL¹⁴) ;
- arts (UFR Arts, philosophie, esthétique) ;
- droit (UFR droit) ;
- économie et gestion (UFR AES¹⁵-EG¹⁶)
- sciences, technologies, santé (UFR MITSIC¹⁷, UFR CC¹⁸).

Par ailleurs, le campus de Saint-Denis renferme, en son sein, quatre écoles doctorales, ainsi qu'un IEE¹⁹ et un IFG²⁰. Certaines formations proposées par ce campus peuvent également être suivies dans le cadre d'un enseignement à distance via son IED²¹. C'est le cas, entre autres, de la licence d'informatique ou encore de celle de droit.

¹⁰ Langues, Littératures et Civilisations Etrangères

¹¹ Langues Etrangères Appliquées

¹² Etudes, Recherche et Ingénierie en Territoires – Environnements - Sociétés

¹³ Sciences de l'Education, Psychanalyse et Communication/Français Langue Etrangère

¹⁴ Sciences Du Langage

¹⁵ Administration Economique et Sociale

¹⁶ Economie et Gestion

¹⁷ cf : note 2

¹⁸ Culture et Communication

¹⁹ Institut d'Etudes Européennes

²⁰ Institut Français de Géopolitique

²¹ Institut d'Enseignement à Distance

L'« université Monde » possède également deux composantes ne se trouvant, cette fois, pas à Saint-Denis :

-l'IUT²² de Montreuil qui propose quatre départements : un département informatique, un département GACO²³, un département QLIO²⁴ et un département information-communication. En plus des quatre DUT²⁵ que celui-ci propose, il est également possible d'y préparer cinq LP²⁶, celle envisagée devant, bien entendu, être compatible avec le DUT précédemment préparé. Les DUT et LP proposés peuvent être préparés, totalement ou partiellement, dans le cadre de l'alternance. 587 étudiants y préparaient un diplôme au cours de l'année universitaire 2015-2016. Il compte, par ailleurs, 36 enseignants titulaires, plus de 115 enseignants vacataires et 18 personnels administratifs et techniques ;

-l'IUT de Tremblay-en-France qui propose trois départements : un département GEA²⁷, un département GIM²⁸ et un département GLT²⁹. En plus des trois DUT proposés, il est possible d'y effectuer la LP qu'elle propose. Ces formations peuvent être préparées totalement ou partiellement en alternance. Près de 300 étudiants y préparent un diplôme chaque année.

B Une ouverture sur le monde permettant une mixité sans égale

Nous parlerons, dans cette partie, des différents partenariats avec différentes universités dans le monde ainsi que des divers horizons des étudiants qui étudient au sein de l'université.

²² Institut Universitaire de Technologie

²³ Gestion Administrative et Commerciale des Organisations

²⁴ Qualité Logistique Industrielle et Organisation

²⁵ Diplôme Universitaire de Technologie

²⁶ Licence Professionnelle

²⁷ Gestion des Entreprises et des Administrations

²⁸ Génie Industriel et Maintenance

²⁹ Gestion Logistique et Transport

1) Un partenariat avec des universités mondialement reconnues

L'« université-monde » est en partenariat avec 237 universités dans le monde, et sur tous les continents, où les étudiants peuvent y effectuer entre 6 mois et un an d'études dans le cadre d'un de leur diplôme, à condition que ceux-ci maîtrisent correctement la langue du pays d'accueil, certaines universités étant très sélectives. On en compte 2 en Océanie, en Australie, 148 en Europe, 12 en Asie, 5 en Afrique et 70 en Amérique (39 en Amérique du Nord, 8 en Amérique Centrale et 23 en Amérique du Sud). Parmi les universités les plus connues avec lesquelles le partenariat est effectif, on inclut l'université Berkeley, en Californie, aux Etats-Unis, l'université de Madrid, ou encore celles de Berlin, de Londres, de Chicoutimi au Québec, de Buenos Aires, de Tokyo, etc... Ces partenariats s'effectuent, notamment, via le programme d'échanges ERASMUS³⁰ pour l'Europe ou CREPUQ³¹ pour le Québec, mais il en existe d'autres pour les autres échanges, notamment le MICEFA³² pour les Etats-Unis. Pour l'Afrique et l'Océanie, des accords bilatéraux sont mis en place, sous forme de documents attestant le droit de partir dans l'université d'accueil dès l'instant que les étudiants sont en règles au niveau de leurs papiers.

³⁰ European Action Scheme for the Mobility of University Students

³¹ Conférence des Recteurs Et des Principaux des Universités du Québec

³² Mission Interuniversitaire de Coordination des Echanges Franco-Américains

2) Des étudiants de tous les horizons étudient au sein de l'université

L'université accueille des étudiants de toutes les nationalités, notamment des étudiants anglophones ou germaniques dans le cadre des formations proposées par l'IEE³³. Par ailleurs, l'UFR MITSIC³⁴ accueille beaucoup d'étudiants africains, aussi bien venant du Maghreb que de l'Afrique subsaharienne.

Par ailleurs, les étudiants ont l'avantage, en postulant à l'université Paris VIII – Vincennes – Saint-Denis, de pouvoir postuler à une ou plusieurs formations qui les intéressent, et ce, quel que soit le baccalauréat ou équivalent (DAEU³⁵ notamment) obtenu. Or, dans les autres universités françaises, bien qu'il soit, en théorie, possible d'accéder à toutes les formations, le seul prérequis étant le baccalauréat, certaines formations sont sélectives, notamment en ce qui concerne la série du baccalauréat de l'étudiant. Ainsi, il est très fortement recommandé, voire indispensable, d'avoir un baccalauréat Scientifique, avec de bonnes notes dans les matières scientifiques, notamment en mathématiques, pour étudier l'informatique à l'université Paris 7 – Diderot. D'ailleurs, pour reprendre l'exemple de l'informatique, l'université Paris 8 accentue ses programmes autour des notions importantes à acquérir afin d'être opérationnel en entreprise, l'informatique théorique y étant peu enseignée et les mathématiques n'étant pas approfondies, contrairement à la plupart des universités où elle est enseignée. De ce fait, chaque étudiant, quel que soit l'horizon, a toutes ses chances de réussir, la motivation étant le critère le plus important pour étudier une discipline au sein de l'université Paris 8.

³³ cf : note 19

³⁴ cf : note 2

³⁵ Diplôme d'Accès aux Etudes Universitaires

Il n'est donc pas rare de voir des étudiants issus de baccalauréats professionnels y étudier, voire des étudiants de BTS³⁶ qui font, ensuite, des études en première année de licence dans une discipline qui n'était pas liée au diplôme précédent.

C Une politique sur le handicap en avance sur d'autres universités

Nous évoquerons, dans cette sous-partie, le service handicap de l'université en présentant ses missions. Nous parlerons également de la position occupée durant ce stage.

1) Présentation de la loi du 11 février 2005, du service et de ses missions

Le service Accueil-Handicap de l'université Paris 8 Vincennes – Saint-Denis a été créé en 2003, deux ans avant la création et l'entrée en vigueur de la loi N° 2005-102 sur l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées. Cette loi a, en effet, été votée et acceptée le 11 février 2005.

Cette loi favorise et facilite l'intégration des personnes handicapées au sein de la société, à commencer par l'intégration des élèves au sein de leur établissement de secteur lorsque le handicap le permet. Elle favorise également l'intégration dans l'enseignement supérieur. Cette loi devait, en théorie, rendre tous les lieux publics accessibles d'ici 2015. Or, nous sommes forcés d'admettre que c'est encore loin d'être le cas.

³⁶ Brevet de Technicien Supérieur

Malgré la rédaction de cette loi, il y a maintenant une douzaine d'années, il est important de noter que 80 % des élèves en situation de handicap n'atteignent pas le niveau du baccalauréat et que les étudiants en situation de handicap ne représentent que 0.4 % de la population étudiante. Par ailleurs, les salariés handicapés ne représentent que 3 % de tous les salariés.

Le service Accueil-Handicap de Paris 8 a pour principale mission d'établir le projet d'accompagnement de chaque étudiant en situation de handicap selon ses besoins, aussi bien pour le déroulement des cours que pour les examens. Les aides sont diverses et peuvent être classées en deux catégories :

- les aides humaines (preneur de notes, secrétaire et/ou surveillant d'examens, interprète en LSF³⁷, etc...). Il est important de noter que le preneur de notes doit suivre exactement les mêmes cours que l'étudiant qui en fait la demande. Concernant le secrétaire d'examen, il doit être neutre en ne faisant que rédiger sous la dictée de l'étudiant, afin de ne pas influencer le raisonnement de l'étudiant, et par le fait, le résultat de la copie ;
- les aides techniques (carte d'accessibilité aux locaux et ascenseurs, clés WC accessibles, agrandissements, transcriptions braille, etc...).

Ces aides sont accordées, suite à la visite médicale annuelle de l'étudiant, par la médecine préventive de l'université. Ces mesures sont ensuite appliquées, à la suite de cette notification, par l'accueil-handicap. Il est important de noter qu'il n'est pas nécessaire d'avoir un dossier MDPH³⁸ pour bénéficier de ces aides, les dispositions d'aménagements étant aussi bien accordées dans le cadre d'un handicap définitif qu'évolutif ou temporaire.

³⁷ Langue des Signes Française

³⁸ Maison Départementale des Personnes Handicapées : organisme créé en 2006 dans chaque département français, afin de remplacer la CDES (Commission Départementale de l'Education Spéciale) pour les enfants et la COTOREP (Commission Technique d'Orientation et de Reclassement Professionnel) pour les adultes par la CDAPH (Commission des Droits et de l'Autonomie des Personnes Handicapées) (cf : documentation technique, partie 4, pour plus de détails sur les aides que peut accorder celle-ci)

L'accueil-handicap peut également aider les étudiants à constituer leur dossier MDPH, les aider à rédiger leur CV et/ou leurs lettres de motivation pour leur recherche d'un stage ou d'un futur emploi et leur donner des liens utiles d'associations pour pouvoir trouver plus facilement ces emplois. C'est le cas notamment de l'association tremplin-handicap.

2) Position occupée au sein du service

L'accueil-handicap de l'université est composé d'une équipe de 7 membres permanents et de 160 vacataires par an (preneurs de notes, secrétaires et surveillants d'examens). Dominique ARCHAMBAULT, chargé de mission handicap, aidé de la présidence, de la DGS³⁹ et de la DGA⁴⁰ Lucie HAZEMANN, est le supérieur hiérarchique de la responsable administrative et financière du service accueil-handicap, Elka PARVANOVA. Celle-ci gère tous les vacataires et dirige Antonella GESBERT, Clément SARGOT et, jusqu'à peu, Myriam SAGERER, tous trois chargés d'accueil et coordination pédagogique.

Durant mon stage, j'étais, au sein du service, sous la direction de Elka PARVANOVA⁴¹ et sous celle de mon référent technique, Vincent BOYER.

Pour conclure cette partie, nous pouvons dire que l'université Paris 8 Vincennes Saint-Denis, de par sa grande histoire, malgré sa relative jeunesse, et son ouverture sur le monde qui l'entoure, est une université différente des autres et, en avance sur son temps, par certains côtés.

³⁹ Direction Générale des Services

⁴⁰ Directrice Générale Adjointe

⁴¹ cf : annexe A p 1

II Un projet de développement très utile et pratique pour l'accueil-handicap

Dans cette seconde partie, l'application à développer ainsi que son fonctionnement seront présentés. Seront ensuite argumentés les choix de programmation ainsi que les contraintes rendant le projet plus difficile et plus long que prévu.

A Présentation de l'application et de son fonctionnement

Il s'agira, dans cette première sous-partie, de présenter l'application que nous avons commencé à développer ainsi que son mode de fonctionnement.

1) Présentation de l'application

Durant mon stage, je devais commencer à développer une application de gestion des informations des étudiants en situation de handicap de l'université Paris 8 Vincennes – Saint-Denis, tous campus confondus. Ce projet de développement a pour but de faciliter la création et la mise à jour des dossiers qui sont, pour le moment, au format papier, ce qui nécessite donc de les classer pour les retrouver facilement ou de les archiver lorsque les étudiants quittent l'université et au cas où ils reviendraient poursuivre des études plus tard. Il arrive, de plus, que certains dossiers ne soient pas retrouvés, ce qui veut dire qu'il faut refaire le dossier, ce qui prend du temps.

Ces informations concernant l'étudiant sont diverses : les informations générales (nom, prénom, date de naissance, adresse, etc...), les projets personnels et professionnels, le type de handicap, les aides et prestations accordées par la MDPH⁴², le parcours antérieur à celui effectué à l'université Paris 8, les inscriptions par années, mises à jour tous les ans, les aménagements des cours et des examens, ainsi que le carnet de visite répertoriant les dates des visites au service ainsi que leurs objets. Or, toutes ces informations sont réparties dans un seul fichier grand fichier Microsoft Excel où chaque ligne de la feuille de calculs représente un étudiant, ce qui rend la mise à jour de ces informations assez difficile, d'autant que l'université compte environ 300 étudiants en situation de handicap, en incluant à la fois le campus de Saint-Denis, mais aussi ceux de Montreuil Sous-Bois et de Tremblay-en-France.

L'idée était donc de proposer une application permettant, après recherche d'un étudiant, d'accéder à ses informations, celles-ci étant réparties sur plusieurs écrans en fonction du type (générales, handicap, aménagements, etc...).

2) Présentation du fonctionnement de l'application

Lorsqu'on démarre l'application, l'utilisateur est invité à rechercher un étudiant en renseignant, au moins l'un des trois champs de saisie, le premier étant le numéro d'étudiant, le second, celui du nom de famille et le troisième et dernier, celui du prénom.

Lorsqu'aucun champ n'est renseigné et que l'utilisateur appuie sur le bouton « Suivant », un message d'erreur indique qu'il faut qu'il renseigne au moins l'un des trois champs de saisie.

⁴² cf : note 38

Lorsque le champ du numéro d'étudiant est renseigné, les deux autres champs ne sont pas éditables. En effet, le numéro d'étudiant étant unique, il n'est pas nécessaire d'inclure le nom et le prénom comme critères de recherche. L'inverse aurait dû être géré de la même manière d'ailleurs, c'est-à-dire que le champ numéro étudiant ne soit pas éditable si au moins l'un des deux autres champs est renseigné, ce qui n'est pas le cas actuellement⁴³.

Le champ numéro étudiant ne doit contenir que des chiffres et indique un message d'erreur dans le cas contraire.

Lorsque le champ nom et/ou le champ prénom est renseigné, un message d'erreur est indiqué si l'étudiant recherché n'est pas répertorié. Si au moins l'un des deux champs nom et prénom est renseigné et que plusieurs étudiants répertoriés ont ce nom et/ou ce prénom en commun, une pop-up indique les étudiants ayant ce point commun. Ceci n'est toutefois que provisoire, le but étant que ce cas montre à l'utilisateur un écran où tous les étudiants ayant ce nom et/ou ce prénom en commun soient écrits avec leurs numéros d'étudiants respectifs et que l'utilisateur pourrait cliquer afin d'accéder à l'écran de l'étudiant recherché.

Lorsque l'étudiant recherché est unique, l'utilisateur arrive sur l'écran d'accueil de l'application. Celui-ci est composé d'un bouton en forme de maison permettant de revenir à l'écran d'authentification, d'un bouton « Ok » qui permettra, par la suite, de passer d'un onglet à l'autre en enregistrant les informations ajoutées ou modifiées, d'un bouton « Annuler » qui réinitialisera l'écran aux valeurs de départ (écran vide dans le cas d'un nouvel étudiant et écran contenant les données de départ dans le cas d'un étudiant existant) et d'onglets représentant les différents écrans où sont affichées les informations relatives à l'étudiant, en fonction de la catégorie (identité, handicap, etc...).

⁴³ cf : documentation technique du projet, annexe E, partie 8

Par ailleurs, l'écran concernant les aménagements est lui-même divisé en deux écrans, lui aussi via un système d'onglets, l'un étant consacré aux aménagements des cours, l'autre à ceux des examens.

Lorsqu'un étudiant n'est pas répertorié, il est possible de créer son dossier en appuyant sur le bouton « Nouveau » de l'écran d'authentification. L'écran d'accueil est alors présenté et chaque écran possède les mêmes champs, vides cette fois, et qu'il faut que l'utilisateur complète. Certains de ces champs, notamment dans l'écran d'identité, regroupant les informations générales de l'étudiant, ne sont pas éditables en mode consultation car, s'il y a des erreurs, ces informations doivent être remontées aux gestionnaires d'Apogé qui corrigent les erreurs, cette application étant prévue, à moyen ou long terme, pour récupérer l'identité des étudiants via cette plate-forme. En revanche, les autres écrans ont leurs champs qui restent éditables même en mode consultation, car ces informations relèvent uniquement de l'accueil-handicap dont l'équipe peut modifier, elle-même, ces informations.

Il est important de noter que la fonctionnalité de suppression d'étudiants n'est pas prise en charge dans la mesure où le numéro d'étudiant est attribué à vie, que l'étudiant soit en situation de handicap ou pas. De fait, si une personne décide de reprendre des études au cours de sa vie, elle reprendra son numéro d'étudiant et son dossier sera mis à jour.

Certains écrans ne font pas apparaître toutes les informations disponibles, celles-ci n'apparaissant que lorsqu'une certaine condition est remplie (bouton radio coché ou case à cocher cochée). C'est le cas, notamment de l'écran des aménagements où, si le bouton radio « Oui » des aides par catégories (humaines ou techniques) est coché, toutes les aides associées apparaissent.

Si la case à cocher « Fait » devant une aide est cochée, la case « Droit » est cochée et n'est pas modifiable à la main afin d'éviter les erreurs. Elle ne devient de nouveau cochable ou décochable que si la case « Fait » est décochée. La case « Droit » indique que l'étudiant a droit à un aménagement mais que celui-ci ne l'utilise pas forcément, contrairement à la case « Fait » où ce droit est appliqué car utilisé⁴⁴.

B Des contraintes précises à respecter rendant le projet plus difficile que prévu

Nous parlerons, dans le cadre de cette partie, du développement initialement prévu ainsi que des contraintes qui nous ont poussé à nous y prendre autrement.

1) Développement initialement prévu

Quand le projet m'a été présenté, j'avais tout de suite une idée précise de programmation. Il s'agissait, pour moi, de développer cette application sous la forme d'un site web au moyen des langages HTML⁴⁵, CSS⁴⁶ et PHP⁴⁷. Il fallait donc que je « transforme » mon ordinateur en serveur, afin de pouvoir faire tourner l'application localement sur mon ordinateur. Je devais, pour cela, installer la plate-forme MAMP⁴⁸ composée des logiciels suivants :

⁴⁴ cf : annexe J p XXVII-XXXIV

⁴⁵ cf : glossaire, annexe B

⁴⁶ cf : glossaire, annexe B

⁴⁷ cf : glossaire, annexe B

⁴⁸ Macintosh Apache MySQL PHP (il s'agit de WAMP pour Windows et de XAMPP pour Linux, le deuxième P signifiant Perl, un autre langage de programmation dont il ne sera pas question ici)

-Apache : logiciel permettant de simuler le comportement d'un serveur ;
-MySQL : SGBD⁴⁹ permettant de créer une ou plusieurs base de données et d'interagir avec elle(s) au moyen de requêtes de sélection, d'ajout, de mise à jour et de suppression. C'est lui qui, par exemple, lors de l'envoi d'un formulaire renseigné par l'utilisateur, exécute la requête afin d'insérer l'utilisateur lorsqu'il n'existe pas ou de le sélectionner pour qu'il accède ensuite à un autre écran. Quand l'utilisateur n'est pas trouvé dans la sélection, la requête ne retourne pas de résultat et le développeur fait en sorte que le programme indique une erreur à l'utilisateur ;

-PHP : c'est le langage de programmation qui permet de rendre un site web dynamique. Les langages HTML et CSS, à eux seuls, ne permettent que de créer des sites au contenu statique, qui ne change jamais. C'est le cas d'un site autobiographique sur quelqu'un par exemple. Il est impossible, avec ces langages, de gérer le traitement des informations contenues dans un formulaire, bien que ceux-ci permettent de le créer, le HTML gérant le contenu (champs de saisie, boutons, photos, vidéos, musiques, etc...), alors que le CSS gère la forme (les couleurs, les bordures, etc...), afin que le site soit le plus ergonomique et donne envie d'être regardé.

Les plates-forme de développement WAMP, MAMP et XAMPP permettent de se connecter localement à PHPMyAdmin, permettant de créer et gérer les informations de la base de données via une interface web graphique et non via un terminal⁵⁰.

⁴⁹ cf : glossaire, annexe B

⁵⁰ Ligne de commande sous forme d'un écran souvent noir (appelé invitée de commandes sous Windows et terminal sous Mac et Linux) et servant à interagir directement avec notre ordinateur sans passer par la souris et par des interfaces graphiques.

La base de données que j'aurai créée, dans le cadre de cette application, aurait consisté en une table⁵¹ pour chaque écran de l'application, contenant les informations relatives à celui-ci. Celles-ci auraient été liée à la table « etudiant » dont le numéro aurait été la clé primaire, c'est-à-dire un identifiant unique par lequel le retrouver et le différencier des autres. Cela permet également de respecter la contrainte d'intégrité des données, afin d'éviter que plusieurs étudiants aient le même numéro, en empêchant d'insérer un autre étudiant avec ce même numéro. Les autres tables auraient eu une référence sur ce numéro d'étudiant, appelée clé étrangère, ce qui aurait permis, pour chaque écran, d'afficher toutes les informations de l'étudiant recherché via l'écran d'authentification.

2) Des difficultés causées par les contraintes empêchant le développement initial

J'ai dû, toutefois, m'y prendre autrement pour développer l'application. En effet, le responsable de la DSI⁵² ainsi que le responsable de l'application Apogé, recensant toutes les informations identitaires et les notes de l'étudiant ne voulaient pas que j'utilise leur base de données pour travailler avec, ni même que j'en crée une moi-même. J'ai donc abandonné l'idée d'utiliser le PHP. Il a donc fallu que je crée des fichiers de données fictifs avec lesquels l'application allait fonctionner. Il a fallu que je crée un fichier de données par écran, chaque fichier contenant tous les numéros d'étudiants.

⁵¹ Regroupement d'informations se rapportant à une entité précise de la vie courante (une personne par exemple)

⁵² Direction des Systèmes d'Information

Par ailleurs, je devais aussi créer les fichiers permettant de remplir des listes déroulantes présentes dans l'application, notamment pour l'écran sur l'enquête ministérielle (MENESR). Les fichiers de données permettant de remplir les champs de saisie et de cocher les boutons en conséquence, dans le cadre de la consultation d'un étudiant, sont au format csv et les fichiers permettant de remplir les listes déroulantes et les listes à choix multiples sont au format txt. Certains fichiers txt serviront aussi à écrire dynamiquement les libellés de certains écrans, comme celui du handicap, où les nominations peuvent changer d'une année à l'autre, bien que ces nominations soient, pour le moment, écrites en brut dans le programme.⁵³

Après cela, il a fallu que je choisisse un autre langage de programmation que le PHP pour réaliser mon application. C'est alors que l'idée d'utiliser le langage orienté objet⁵⁴ java m'a paru une évidence.

C Le langage java : un choix stratégique par rapport à d'autres langages

Nous justifierons, ici, nos choix de programmation pour ce projet.

1) Un langage multiplateforme très utilisé sur le marché du travail.

Le langage de programmation Java, créé en 1995, est un des langages les plus utilisés sur le marché du travail.

⁵³ cf : annexes K et L

⁵⁴ cf : glossaire, annexe B

Il présente l'avantage d'être portable sur toutes les plates-forme (Windows, Mac ou Linux). Cela est rendu possible par le fait que le compilateur⁵⁵ java soit intégré dans une machine virtuelle qui transforme le code écrit par le développeur en un code semi-compilé encore non lisible et exécutable par un ordinateur. Ce code semi-compilé, appelé aussi bytecode, sera transformé en code binaire lors de l'exécution (lancement) du programme par la machine virtuelle. Ainsi, un code écrit en java sur une machine équipé d'un système d'exploitation Windows sera exécutable sur Linux et Mac, et inversement. Or, ceci représente l'avantage de ne pas être obligé de recompiler le code d'une machine à l'autre, contrairement à d'autres langages tels que le C ou le C++ qui sont des langages compilés⁵⁶ entièrement en code natif. De fait, un programme écrit avec un de ces langages doit être recompilé à chaque fois par la nouvelle machine avant de pouvoir être exécuté.

Bien que, dans ces débuts, le langage java souffrait de problèmes liés aux performances d'exécution des programmes, liés eux-mêmes par cette machine virtuelle intermédiaire, de gros progrès ont aujourd'hui été réalisés. Il n'en reste pas moins que, pour des programmes devant privilégier les performances à la portabilité⁵⁷, le langage java n'est pas adapté, à moins d'utiliser un compilateur tel que GCJ⁵⁸.

⁵⁵ cf : glossaire, annexe B

⁵⁶ Transformés en code binaire exécutable par un ordinateur en fonction de son architecture interne

⁵⁷ Le fait qu'un programme soit exécutable sur toutes les machines sans être obligé de recompiler le code au préalable pour l'architecture donnée.

⁵⁸ GNU (GNU is Not UNIX) Compiler for Java : compilateur open source (non propriétaire et dont le code est accessible et modifiable par tous) permettant de compiler entièrement du code java en code natif sans passer par l'intermédiaire de la Java Virtual Machine (JVM).

Le langage Java présente l'avantage, par rapport à d'autres langages de programmation, d'inclure dans ses composants de base, contenus dans le JDK⁵⁹, des librairies permettant de créer, de manière relativement simple, des interfaces graphiques. Ceci est même rendu encore plus simple en utilisant un plug-in⁶⁰ inclus dans l'IDE⁶¹ Eclipse s'appelant WindowBuilder. Il permet de créer rapidement les interfaces graphiques en positionnant les composants que l'on veut utiliser directement sur la fenêtre prévue à cet effet. Le code java permettant de créer ces composants et de les positionner est automatiquement généré. C'est la méthode du WYSIWYG⁶².

Il permet aussi, au même titre que le PHP⁶³, de créer des applications sous forme d'applets⁶⁴, au moyen des pages JSP⁶⁵. Il faut, pour ce genre de développement, utiliser la version J2E⁶⁶ de java. Par ailleurs, le JDK contient des librairies permettant de se connecter et d'interagir avec les bases de données, à condition d'inclure le JDBC⁶⁷ adéquat au SGBD⁶⁸ utilisé, que l'on peut trouver sur Internet, sous forme d'un fichier jar⁶⁹.

Enfin, la plate-forme apogé étant développée, elle-même en java, il est stratégiquement plus judicieux d'utiliser ce même langage afin d'assurer la compatibilité des composants sur le long terme, une fois que le projet sera plus abouti.

⁵⁹ Java Developpement Kit (kit de développement java)

⁶⁰ Extension d'un logiciel

⁶¹ cf : note 3

⁶² What You See Is What You Get (ou Ce que vous voyez est ce que vous obtenez)

⁶³ cf : note 47

⁶⁴ cf : glossaire, annexe B

⁶⁵ cf : glossaire, annexe B

⁶⁶ Java Enterprise Edition

⁶⁷ cf : glossaire, annexe B

⁶⁸ cf : note 49

⁶⁹ Java Archive : fichier ZIP (compressé) pour utiliser un certain nombre de classes java.

2) Un paradigme de programmation rendu relativement simple par le langage : la POO⁷⁰

La programmation orientée objets est un paradigme de programmation cherchant à représenter, au maximum, des entités de la vie réelle. Ces entités sont représentés par des objets, créés via une classe⁷¹ qui est un fichier de code définissant les caractéristiques d'un objet, appelées attributs ainsi que les comportements que celui-ci doit adopter, appelés méthodes. Les constructeurs sont des méthodes un peu particulières permettant de créer des instances de ce type d'objets. Ils peuvent être vus comme des moules dont les objets prennent la forme une fois créés.

Le paradigme orienté objet dispose de méthodes permettant, encore plus que les paradigmes fonctionnel (comme pour le langage Ocaml) et impératif (comme pour le langage C), de privilégier le refactoring⁷². Cette méthode est utile pour éviter les erreurs de programmation engendrée lorsqu'un même bout de code est écrit à plusieurs endroits. Les erreurs viennent souvent du fait que, lorsque le développeur doit corriger une erreur dans le code, il oublie de le faire à un endroit, ce qui peut engendrer d'autres erreurs qui peuvent ensuite être difficiles à déboguer⁷³. Les deux méthodes principalement connues dans le paradigme orienté objet, afin d'éviter les répétitions de code sont les suivantes :

⁷⁰ Programmation Orientée Objets

⁷¹ cf : glossaire, annexe B

⁷² cf : glossaire, annexe B

⁷³ Enlever un bug au moyen du débogueur inclus dans l'IDE. Ce programme permet d'exécuter le programme pas à pas afin de voir ce qu'il se passe en mémoire à un instant précis et ce que l'ordinateur exécute comme instruction, afin de détecter l'endroit où il y a un problème avant de tenter de le résoudre.

-l'héritage, qui consiste à créer une classe qui hérite de toute ou partie d'une autre classe, en fonction de l'encapsulation⁷⁴ de ses attributs et de ses méthodes. Le mot clé « private » devant un attribut ou une méthode indique que celui-ci ou celle-ci n'est accessible qu'en interne dans la classe. Quand il n'y a pas de mot, l'attribut ou la méthode est accessible à partir de toutes les classes faisant partie du même package⁷⁵ que la classe dans laquelle l'attribut ou la méthode est déclaré(e) comme tel(le). Le mot clé « protected » indique une accessibilité à partir de toutes les classes appartenant au même package ainsi qu'aux classes filles (classes en étendant une autre), même si elles ne se trouvent pas dans le même package. Enfin, le mot clé « public » indique une accessibilité de partout dans le code, celui-ci n'étant utilisé quasiment que pour les méthodes que l'on a besoin d'appeler après la création d'un objet⁷⁶ dans une autre classe.

Les classes étendues par d'autres ou en étendant une autre peuvent être déclarées « abstract », c'est-à-dire que certaines de leurs méthodes, si elles sont abstraites, ne sont pas implémentées dans la classe mais dans celles qui l'étendront, à moins d'être elles-mêmes abstraites.

Le langage java, contrairement au C++, autre langage orienté objet, n'autorise l'héritage que d'une seule classe par une autre. En effet, l'héritage multiple peut causer des ambiguïtés si deux classes étendues par une même classe possèdent une méthode ayant une signature⁷⁷ commune mais n'ayant pas le même comportement, car il est impossible de savoir, pour l'ordinateur, la méthode de quelle classe va être appelée. Pour pallier à ce problème, les interfaces ont été introduites dans le langage java ;

⁷⁴ cf : glossaire, annexe B

⁷⁵ cf : glossaire, annexe B

⁷⁶ cf : glossaire, annexe B

⁷⁷ Nom d'une méthode avec ses paramètres éventuels entre parenthèses

-les interfaces, qui peuvent être vues comme des classes entièrement abstraites dont les méthodes, bien qu'ayant le même nom, n'ont pas le même comportement en fonction de la classe qui va implémenter la ou les interfaces en question. Effectivement, à l'inverse de l'héritage, le nombre d'interfaces implémentées par une classe en java est illimité.

Enfin, le paradigme objet permet d'utiliser des patterns⁷⁸ de programmation permettant une conception plus claire et concise du programme. L'exemple le plus simple, utilisé d'ailleurs dans ce projet, est le pattern singleton, consistant à créer un constructeur « private » et donc uniquement accessible à partir de la classe et de créer, dans cette même classe, une méthode « public static » afin qu'elle soit accessible de n'importe où dans le code et sans être obligé d'instancier (créer) un objet pour accéder à cette méthode. Il sert, dans le cas du projet, à ne créer qu'une seule instance de la classe Fenetre qui est retournée si elle existe, et créée sinon, afin de pouvoir modifier son contenu en y passant en paramètre l'écran que l'on veut y afficher. On s'en sert également dans le cas d'une connexion à une base de données, afin de ne s'assurer qu'une seule connexion a lieu.

Pour finir cette partie, nous pouvons affirmer que le développement de ce projet, bien que très pratique et utile à l'accueil-handicap n'est pas sans difficulté, tout comme les autres projets d'ailleurs, d'autant que des contraintes précises étaient à respecter. Enfin, la plate-forme apogé ayant été développée en java, il a finalement été plus judicieux de se tourner vers ce langage pour ne pas engendrer plus de difficultés à cause de la compatibilité des composants.

⁷⁸ patrons de conception

III Un stage qui se différencie des précédents

Dans cette ultime partie, il sera question d'expliquer en quoi ce stage se différencie des précédents en expliquant que, malgré une gestion plus fluide et libre du temps, une grande autonomie a été nécessaire pour mener à bien ce stage. Il sera également question des documentations et outils utilisés pour les potentiels futurs développeurs qui reprendraient le projet plus tard.

A Un mode de travail très différent...

Nous allons présenter, dans le cadre de cette sous-partie, les méthodes de travail pour accomplir ce stage.

1) Un stage réalisé à partir de rien, grâce aux acquis des stages précédents

Lors de mes précédents stages, ce que je devais développer dépendait d'une application déjà existante, et tout particulièrement lors de mon premier stage effectué dans le cadre de ma deuxième année de DUT informatique. Celui-ci consistait à ajouter des fonctionnalités à une application interne de la Société Générale. Or, je ne savais pas comment m'y prendre. Du coup, j'ai eu tendance à coder sans comprendre le code sur lequel je travaillais, étant donné qu'à l'université, les projets sont souvent à réaliser du début jusqu'à la fin. De fait, la compréhension d'un programme avant de soi-même coder n'était pas du tout apprise. Pourtant, c'est l'une des choses les plus importantes dans le processus de développement, afin d'être sûr, par la suite, de ne pas rendre le projet complètement non fonctionnel. Ce n'est, toutefois, pas toujours évident, dans la mesure où chaque développeur a ses propres manières de coder.

De plus, quand je n'arrivais pas à mes fins durant le codage d'une fonctionnalité, je passais à une autre dans l'espoir que cette nouvelle fonctionnalité soit plus simple à développer. Or, faire cela peut entraîner plus de bugs⁷⁹ qu'autre chose.

Lors de mon stage de licence, au sein du groupe Generali, je devais développer une interface d'administration de la base de données d'une application. Celle-ci a été développée indépendamment de l'application en question mais devait, une fois le développement terminé, y être rattachée. Durant ce stage, je n'ai pas reproduit les erreurs effectuées lors du stage précédent et j'ai davantage appris à travailler en autonomie, en apprenant trois Frameworks⁸⁰ beaucoup utilisés dans le monde du travail : GWT⁸¹, qui permet de créer des interfaces graphiques en java et traduites en langage javaScript, qui est un langage du web, Hibernate, pour gérer les requêtes SQL en encapsulant les données de la base directement dans la classe représentant la table dans la base de données, et Spring, permettant l'instanciation automatique d'objets java.

J'ai également appris, pendant ce stage de licence, à faire le point au travers de réunions quotidiennes (stand-up), avec l'équipe de développement. Or, j'ai été amené à faire le point, de temps en temps, avec l'équipe pour discuter de ce qui était bien fait et de ce qui était à refaire, notamment au niveau des écrans de l'application.

Durant mes deux autres stages, j'étais encadré par un informaticien qui m'aidait à accomplir mes tâches en me donnant les conseils adéquats.

⁷⁹ Fonctionnement anormal dans l'exécution d'un programme qui doit, selon le cas, être corrigé le plus rapidement possible.

⁸⁰ Cf : glossaire, annexe B

⁸¹ Google Web Toolkit

C'est d'ailleurs, lors du stage chez Generali que j'ai appris à me servir du débogueur et à comprendre sa réelle utilité. J'y ai également appris l'utilité du refactoring⁸², philosophie que j'ai beaucoup utilisée au cours de mon actuel stage.

J'ai enfin appris, au cours de ces deux stages précédents, à faire des sauvegardes régulières du code lorsque celui-ci s'avérait fonctionnel afin de pouvoir revenir à une version antérieure si la version en cours de développement s'avérait non fonctionnelle avec un retour au point de départ difficile. Il est, en effet, parfois difficile, même avec le débogueur, de comprendre d'où vient le problème que l'on a introduit en développant. Je l'ai, d'ailleurs, appliqué cette année, ayant eu un cours approfondi là-dessus en début d'année, alors que j'entamais mon Master.

2) La découverte du télétravail

Effectuant mon actuel stage au sein même de l'université, j'étais donc régi par les règles de fermeture de celle-ci au mois d'août, même si ce stage se déroulait jusqu'au 31 août inclus. Ainsi, le fait de développer le projet en cours, indépendamment d'Apogé s'est révélé pratique, dans la mesure où, pendant trois semaines, j'ai pu travailler à distance à partir de chez moi. De fait, j'étais beaucoup plus libre dans la manière de gérer mon temps, n'ayant pas de transports à prendre et pouvant faire des pauses plus longues qu'en entreprise dans la journée. En contrepartie, je travaillais parfois tard le soir et pouvais m'arranger en travaillant plus pour prévoir quelque chose à un autre moment, où je ne pouvais pas travailler en même temps, et je dois dire que c'est une situation qui m'a beaucoup plu.

⁸² cf : glossaire, annexe B

Cette situation s'est toutefois révélée piégeuse dans la mesure où, à mesure que les développements avançaient et s'avéraient fonctionnels, je n'arrivais pas à m'arrêter, et reculais sans cesse la rédaction du rapport, au profit de l'avancement du projet.

B ...Nécessitant une grande autonomie

Nous allons, ici, comparer la méthode de travail utilisée dans le cadre de ce stage à celle utilisée dans le cadre de travaux universitaires.

1) Un stage réalisé grâce à des pistes...

Ce stage très intéressant, de par sa nature qui m'était inconnue, nécessitait une grande autonomie. En effet, j'ai dû commencer à développer mon projet à partir de zéro, en m'inspirant seulement d'une maquette fournie par l'accueil handicap. De plus, je n'avais pas un informaticien pour m'encadrer au moindre blocage dans la programmation, contrairement aux deux autres stages. Toutefois, c'est un mal pour un bien, dans la mesure où quand ce que j'entreprenais fonctionnait, j'appréciais davantage, du fait que je trouvais moi-même la solution à mon problème. C'est notamment le cas quand je devais développer les écrans de l'application. Ceux-ci, initialement étaient intégrés à la classe Fenetre⁸³, ce qui, au fur et à mesure, alourdissait la classe, ce qui risquait, par la suite, de rendre le code de plus en plus difficile à lire et à maintenir. J'ai donc cherché une solution qui me permettrait de séparer mes écrans de ma fenêtre.

⁸³ cf : annexe C

Puis, j'ai pensé au pattern singleton qui s'est avéré fonctionnel et qui devait impérativement être utilisé, le contenu de la fenêtre ne pouvant être accessible à partir des écrans, qui étendaient la classe JPanel⁸⁴. Il fallait donc, impérativement que j'accède à la fenêtre pour modifier directement son contenu via la méthode setContentPane, prenant en paramètre le JPanel étant l'écran à afficher, ce qui m'aurait obligé, si je n'avais pas adopté la méthode actuellement utilisée, à créer plusieurs instances de la fenêtre, ce qui m'en aurait ouvert plusieurs lors du lancement de l'application, ce que je ne voulais pas, bien entendu⁸⁵.

2) ...Tel un projet d'université

Lors des réunions avec Vincent Boyer, mon responsable technique, et l'équipe de l'accueil-handicap, les développements réalisés étaient discutés afin de savoir si les écrans convenaient, dans la mesure où le logiciel allait être utilisé au sein du service. L'équipe jouait donc le rôle des clients qui sont « rois » et dont le développement doit répondre à leurs attentes. Il m'est donc arrivé de devoir totalement retravailler certains fichiers de données et certains écrans, suite aux modifications demandées, notamment concernant l'écran d'aménagements des cours et des examens, celui-ci devant être, initialement, composé de cases à cocher face aux aides proposées. Or, l'équipe a finalement voulu deux cases par aide, l'une indiquant le droit, l'autre indiquant que celui-ci est appliqué. De plus, les aménagements des cours et des examens faisaient, initialement partie du même écran, avant d'être séparés en deux écrans sous forme d'onglets, au sein même de l'onglet aménagements.

⁸⁴ Classe java représentant les écrans qui peuvent s'afficher dans une fenêtre.

⁸⁵ cf : annexe D

Par ailleurs, j'ai dû faire le point seul à seul avec mon chef technique afin de pouvoir discuter des problèmes, notamment celui des barres de défilement que nous avons, tous les deux, tentés de résoudre, sans succès pour le moment. Or, lors des projets tuteurés, nous faisions le point, en binôme avec notre tuteur qui nous donnait des pistes mais il n'était pas derrière nous pour nous dire exactement quoi faire, ce qui est complètement l'inverse lors d'un stage en entreprise.

C L'après stage : un code qui se veut le plus clair et lisible possible

Nous allons, dans cette ultime sous-partie, expliquer ce qui a été fait, afin de rendre le plus facile possible, la reprise du projet par d'autres potentiels développeurs plus tard.

1) Deux documentations techniques accompagnent le projet.

Dans un souci pour les potentiels futurs développeurs qui reprendraient le projet derrière moi, il m'a semblé bon de rédiger une documentation technique assez détaillée du projet, décrivant notamment, en plus de l'explication du pourquoi du développement du projet, ou encore de la technique de développement et de l'architecture du projet, de décrire le fonctionnement de la MDPH concernant les règles d'attributions des aides et/ou prestations aux personnes handicapées, afin que ceux-ci puissent reprendre l'écran MENESR afin de pouvoir le développer correctement.

J'y ai également évoqué les axes d'amélioration possible, notamment la résolution du bug des barres de défilement, qui empêchent de voir l'écran dans sa totalité, ce qui pose problème, dans la mesure où tout ne peut pas être sélectionné et/ou renseigné.⁸⁶

⁸⁶ cf : annexe E

A côté de cette documentation technique, j'ai également rédigé toute la javadoc du projet. Celle-ci consiste en des commentaires particuliers se trouvant dans les classes et qui sont générés, à partir d'un fichier spécial et l'exécutable javadoc contenu dans le JDK, afin de créer une documentation détaillée sous forme de site web. Elle permet aux développeurs de comprendre comment une méthode écrite fonctionne ainsi que ce qu'elle prend en paramètres et ce qu'elle retourne. Celle-ci peut être facilement modifiée et regénérée.⁸⁷

2) Un code très, voire trop, concis

Dans ce même souci de clarté pour ceux qui reprendraient le projet, l'idée était de rendre le code le plus lisible et concis possible au moyen du refactoring. C'est ainsi que j'ai créé, notamment, une méthode qui gère les emplacements et le comportement des composants dans les écrans de l'application, afin que le code de chaque écran soit le moins long possible.

C'est aussi pour régler des cas particulier que j'ai été amené à créer des classes étendant les composants graphiques sans rien ajouter de plus, afin de pouvoir créer des conditions dans la méthode contenue dans cette classe gérant les composants, même si cela a parfois entraîné quelques bugs, comme par exemple, des champs ou des cases à cocher qui ne s'affichaient pas même si un bouton radio « Oui » était coché. Il a donc fallu que j'utilise le débogueur pour comprendre qu'il fallait tout simplement que je déplace mes lignes de codes pour corriger ceux-ci, et tout fonctionne correctement de ce point de vue-là, à l'heure actuelle⁸⁸.

⁸⁷ cf : annexe F

⁸⁸ cf annexe D, photos de la classe abstractJPanel

Pour terminer cette ultime partie, il est bon de souligner que ce stage, bien qu'assez différent des précédents, s'est révélé très intéressant et m'a davantage permis de développer mon autonomie, notamment au travers de la découverte du télétravail et du relativement peu de points faits avec l'encadrant technique, contrairement à un stage en entreprise classique. Il m'a également appris à penser davantage aux autres, sachant que je ne serai pas le seul développeur sur ce projet.

Conclusion

Pour conclure, je dois dire que cette mission de stage a été très intéressante et très constructive, aussi bien sur le plan technique qu'environnemental. Elle m'a, en effet, permis de mettre en pratique mes acquis de mes précédents stages et de découvrir un mode de travail encore inconnu et parfois très pratique.

Bien que la mission n'ait pas été remplie dans sa totalité, du fait de la grande quantité de temps qu'il faudrait pour arriver au terme du projet, celle-ci fonctionne globalement bien, malgré quelques bugs⁸⁹, grâce à une politique basée sur un refactoring parfois extrême, qui pourrait toutefois être encore amélioré. Cette politique a été adoptée en vue de laisser, aux potentiels futurs développeurs qui reprendraient le projet, un code le plus facilement lisible et maintenable.

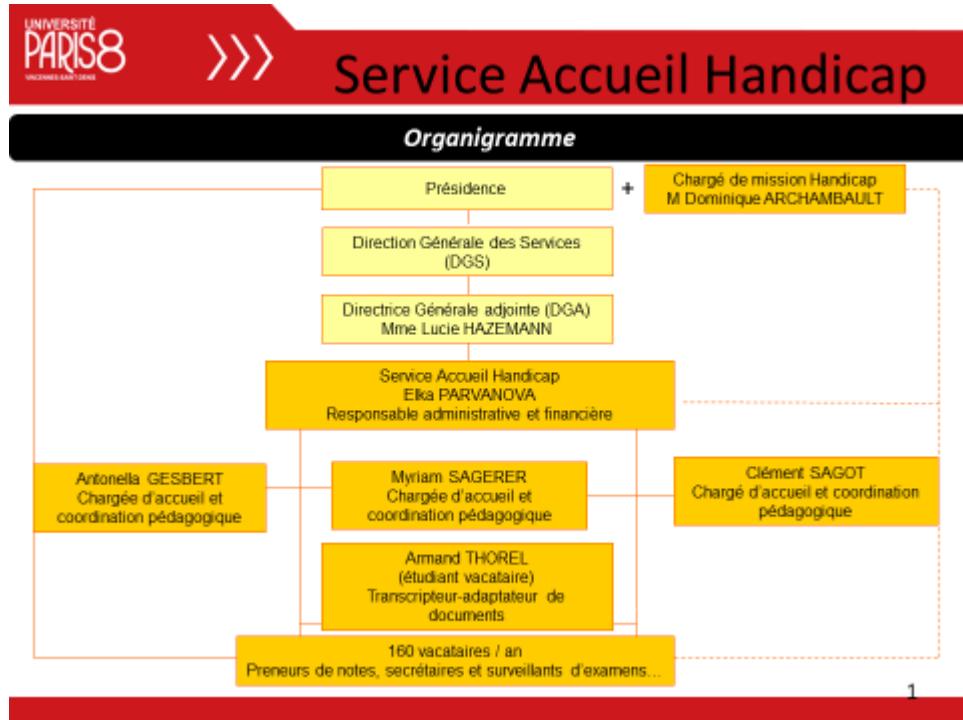
Un axe d'amélioration notable, pour les développements futurs, serait d'utiliser la librairie java Sonar, permettant, selon des règles préalablement établies, de vérifier le code source dans sa totalité. Exécutable, sous forme de page web est alors créé et indique les erreurs que le code peut contenir. Ces erreurs sont de plusieurs types : majeures (le code ne peut pas aller en phase de production et elles doivent être impérativement corrigées), les erreurs mineures (qu'il faut corriger quand on peut, même si elles n'empêchent pas le passage en phase de production). Certaines erreurs détectées, à cause de ces règles établies peuvent être supprimées en utilisant le commentaire NOSONAR. C'est le cas, notamment, quand certaines classes contiennent un trop grand nombre d'attributs (alors que c'est nécessaire) ou quand une méthode devant retourner quelque chose a un trop grand nombre de « return »⁹⁰ (alors que c'est parfois nécessaire également).

⁸⁹ cf : annexe I

⁹⁰ Mot clé de la programmation indiquant que le morceau de code a fini d'être exécuté. La chose retournée peut être un nombre, une chaîne de caractères, voire, comme dans le cas du langage java, un objet.

ANNEXES

Annexe A : Organigramme du service



Annexe B : Glossaire des termes techniques

A

API : Application Programming Interface (ou Interface de Programmation en français) : Ensemble de classes et de méthodes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

Applet : logiciel s'exécutant dans la fenêtre d'un navigateur web

B

Base de données : Ensemble de données référençant un thème commun rangées de manière structurée et pouvant être reliées entre elles

C

Classe : fichier de code utilisé, dans le cadre de la programmation orientée objets, permettant de créer le « moule » de l'objet afin de pouvoir en créer à l'intérieur de cette même classe ou dans d'autres. Elle est constituée d'attributs qui sont les caractéristiques de l'objet et de méthodes qui sont les comportements que doit avoir l'objet.

Compilateur : programme qui est chargé de transformer le code source (code développé par le programmeur) en langage machine (code binaire). En fonction des langages, ce code peut être directement exécuté (comme le C) ou exécuté au moyen d'une machine virtuelle (comme le Java avec la JVM (Java Virtual Machine))

CSS : Cascading Style Sheets : langage de mise en forme de sites web (couleurs, bordures, etc...)

E

Encapsulation : Méthode de la programmation orientée objet permettant d'assurer l'intégrité des données d'une classe.

F

Framework : Bibliothèque logicielle codée dans un langage permettant

de faciliter les développements d'un logiciel, dans ce même langage.
Cela nécessite, toutefois, de comprendre le fonctionnement de celui-ci au préalable.

H

HTML : HyperText Markup Language : langage de gestion de contenu de sites web (texte, formulaires, images, etc...)

I

Integrated Developement Environment (Environnement Développement Intégré) : logiciel permettant la simplification des corrections de bugs (erreurs de compilation ou d'exécution) au moyen d'une compilation en temps réel et d'un débogueur intégré permettant d'exécuter l'application pas à pas

J

JDBC : Java DataBase Connection (permet à un programme java de se connecter et d'interagir avec une base de données).

JSP : Java Server Page : technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique.

L

Langage orienté objet : langage permettant de créer des types de variables représentant des entités du monde réel qui interagissent entre elles (par instanciation et appel des méthodes) pour faire fonctionner le programme créé.

O

Objet : brique logicielle qui possède sa propre existence avec des propriétés définies ainsi que des méthodes définissant les comportements que celui-ci doit adopter.

P

Package : ensemble de classes contenu dans une partie d'une API

PHP : Personnal Home Page : langage de programmation permettant de créer des sites web dynamiques (qui changent en fonction des données fournies pour être affichées par exemple). Il est utilisé pour le développement du réseau social Facebook par notamment. Il permet également des interactions avec une ou plusieurs bases de données.

R

Refactoring : Méthode de programmation consistant à diviser, le plus possible, le code en méthodes afin d'éviter au maximum les répétitions.

S

SGBD : Serveur de Gestion de Bases de Données : serveur chargé d'exécuter les requêtes SQL de l'utilisateur dans une base de données

SQL : Structured Query Language : langage de requêtes pour les bases de données

T

Test : Procédure de vérification partielle d'un système dont l'objectif est d'identifier un nombre maximum de comportements problématiques du logiciel afin d'en augmenter la qualité. Les tests se répartissent en quatre types : -les tests unitaires : on teste les comportements d'une seule partie du logiciel à la fois ; -les tests d'intégration : on teste le logiciel dans son ensemble ;

-les tests système : on évalue ici la conformité aux exigences spécifiées du système ; -les tests d'acceptation (ou recettes) : on s'assure que le produit est conforme aux spécifications

Annexe C : Classe Fenetre.java au début du développement

```
41
42 public class Fenetre extends JFrame {
43
44     private Authentification authentification;
45     private Accueil accueil;
46
47     private Dimension screenSize;
48
49     private int num_etudiant;
50     private ArrayList<String[]> donneesEtudiantsAvecChainesCommunes;
51
52     /**
53      * Create the frame.
54     */
55     public Fenetre() {
56         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57         screenSize=Toolkit.getDefaultToolkit().getScreenSize();
58         screenSize.setSize(1000, 500);
59         setBounds(0, 0, screenSize.width, screenSize.height);
60         this.authentification = new Authentification();
61         setContentPane(authentification);
62         this.authentification.setVisible(true);
63         setVisible(true);
64     }
65
66     private class Authentification extends JPanel {
67         private JTextField textField_num_etudiant;
68         private JTextField textField_nom;
69         private JTextField textField_prenom;
70
71         private JButton btnSuivant;
72
73         private JButton btnNouveau;
74
75         private LectureFichier lectureFichier;
76
77         private Accueil accueil;
78     }
}
```

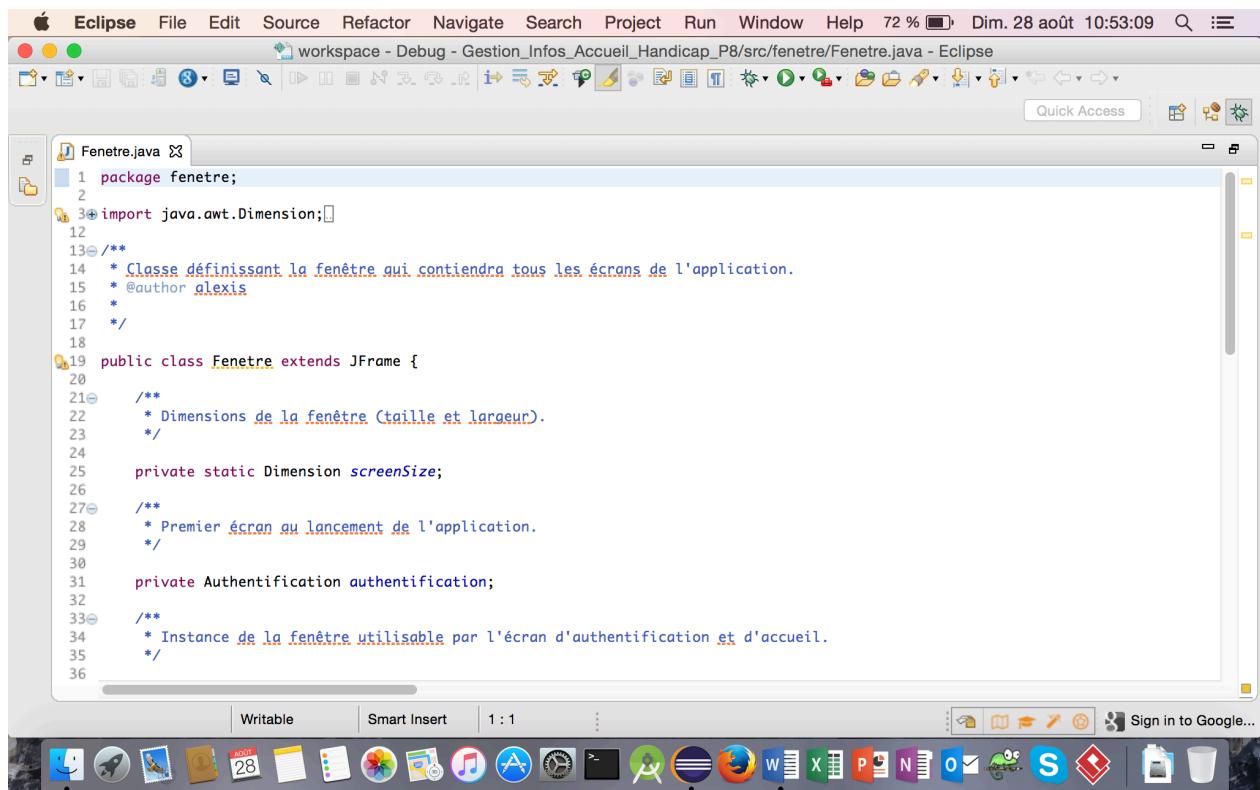
The screenshot shows a Java code editor with the file 'Fenetre.java' open. The code defines a class 'Identite' that extends 'JPanel'. It contains private fields for 20 JDateChoosers and 20 JTextFields, a boolean 'nouveau', and a string 'numEtudiant'. The constructor initializes a JLabel and adds it to the panel. The code is annotated with line numbers from 349 to 386.

```
349     private class Identite extends JPanel {
350
351         private JDateChooser textField;
352         private JDateChooser textField_1;
353         private JTextField textField_2;
354         private JTextField textField_3;
355         private JTextField textField_4;
356         private JTextField textField_5;
357         private JDateChooser textField_6;
358         private JTextField textField_7;
359         private JTextField textField_8;
360         private JTextField textField_9;
361         private JTextField textField_10;
362         private JTextField textField_11;
363         private JTextField textField_12;
364         private JTextField textField_13;
365         private JTextField textField_14;
366         private JTextField textField_15;
367         private JTextField textField_16;
368         private JTextField textField_17;
369         private JTextField textField_18;
370         private JTextField textField_19;
371         private JTextField textField_20;
372
372     private boolean nouveau;
373
374     private String numEtudiant;
375
376     public Identite() {
377         this.nouveau=true;
378
379         setLayout(null);
380
381         JLabel lblNewLabel = new JLabel("Date de création du dossier :");
382         lblNewLabel.setBounds(10, 11, 190, 14);
383         add(lblNewLabel);
384
385         textField = new JDateChooser();
386     }
386 }
```

The screenshot shows a Java code editor window with the file `Fenetre.java` open. The code is a class definition for `Identite` with various GUI components like labels, date choosers, and text fields. The editor has a dark theme with syntax highlighting. A vertical bar on the right side displays code completion suggestions for the current word.

```
377 public Identite() {
378     this.nouveau=true;
379
380     setLayout(null);
381
382     JLabel lblNewLabel = new JLabel("Date de création du dossier :");
383     lblNewLabel.setBounds(10, 11, 190, 14);
384     add(lblNewLabel);
385
386     textField = new JDateChooser();
387     textField.setBounds(204, 8, 150, 20);
388     textField.getDateEditor().setEnabled(false);
389     add(textField);
390
391     JLabel lblNewLabel_1 = new JLabel("Dernière mise à jour :");
392     lblNewLabel_1.setBounds(10, 36, 150, 14);
393     add(lblNewLabel_1);
394
395     textField_1 = new JDateChooser();
396     textField_1.setBounds(204, 39, 150, 20);
397     textField_1.getDateEditor().setEnabled(false);
398     add(textField_1);
399
400     JLabel lblNewLabel_2 = new JLabel("N° Etudiant :");
401     lblNewLabel_2.setBounds(10, 73, 90, 14);
402     add(lblNewLabel_2);
403
404     textField_2 = new JTextField();
405     textField_2.setBounds(112, 70, 218, 20);
406     add(textField_2);
407     textField_2.setColumns(10);
408
409     ButtonGroup monsieurMadame = new ButtonGroup();
410
411     JRadioButton rdbtnM = new JRadioButton("M");
412     rdbtnM.setBounds(10, 103, 50, 23);
413     add(rdbtnM);
```

Annexe D : Classe Fenetre.java modifiée par la suite + quelques classes pour montrer l'évolution du code :



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Eclipse File Edit Source Refactor Navigate Search Project Run Window Help 72 % Dim. 28 août 10:53:09
- Project Bar:** workspace - Debug - Gestion_Infos_Accueil_Handicap_P8/src/fenetre/Fenetre.java - Eclipse
- Toolbar:** Standard Eclipse toolbar icons.
- Left Margin:** Shows line numbers from 1 to 36.
- Code Editor:** The Fenetre.java file content is displayed:

```
1 package fenetre;
2
3 import java.awt.Dimension;
4
5 /**
6  * Classe définissant la fenêtre qui contiendra tous les écrans de l'application.
7  * @author alexis
8  */
9 public class Fenetre extends JFrame {
10
11     /**
12      * Dimensions de la fenêtre (taille et largeur).
13      */
14     private static Dimension screenSize;
15
16     /**
17      * Premier écran au lancement de l'application.
18      */
19     private Authentication authentification;
20
21     /**
22      * Instance de la fenêtre utilisable par l'écran d'authentification et d'accueil.
23      */
24 }
```
- Bottom Bar:** Shows various application icons and a "Sign in to Google..." button.

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help 72 % Dim. 28 août 10 53 18 workspace - Debug - Gestion_Infos_Accueil_Handicap_P8/src/fenetre/Fenetre.java - Eclipse

```
32
33 /**
34  * Instance de la fenêtre utilisable par l'écran d'authentification et d'accueil.
35 */
36
37 private static Fenetre fenetre;
38
39 /**
40  * Create the frame.
41  * @throws NullPointerException
42  * @throws LongueurDifferenteListesException
43  * @see Main
44  * Définit la taille de la fenêtre et instancie le premier écran de celle-ci lors du lancement de l'application (authentification).
45  * Le constructeur est volontairement privée afin d'utiliser le pattern Singleton permettant de n'instancier qu'une seule fois la fenêtre.
46 */
47 private Fenetre() throws LongueurDifferenteListesException, NullPointerException {
48     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
49     screenSize=Toolkit.getDefaultToolkit().getScreenSize();
50     //screenSize.setSize(1000, 500);
51     setBounds(0, 0, screenSize.width, screenSize.height);
52     authentication = new Authentification();
53     setContentPane(authentification);
54     authentication.setVisible(true);
55     /*JScrollPane scrollpane = new JScrollPane(this);
56     scrollpane.setBounds(this.getBounds());
57     add(scrollpane);*/
```

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help 72 % Dim. 28 août 10 53 28 workspace - Debug - Gestion_Infos_Accueil_Handicap_P8/src/fenetre/Fenetre.java - Eclipse

```
57     scrollpane.setBounds(this.getBounds());
58     add(scrollpane);*/
59 }
60 /**
61  *
62  * @return une instance de la fenêtre si elle n'est pas créée ou l'instance courante si elle existe.
63  * @throws LongueurDifferenteListesException
64  * @throws NullPointerException
65  * @see Main
66  */
67
68 public static Fenetre getInstance() throws LongueurDifferenteListesException, NullPointerException {
69     if(fenetre==null) {
70         fenetre = new Fenetre();
71     }
72     return fenetre;
73 }
74
75 /**
76  *
77  * @return les dimensions de la fenêtre afin de pouvoir modifier les valeurs brutes de positionnement des composants dans les panneaux.
78  */
79
80 public static Dimension getScreenSize() {
81     return screenSize;
82 }
83 }
```

The screenshot shows the Eclipse IDE interface on a Mac OS X desktop. The title bar reads "workspace - Debug - Gestion_Infos_Accueil_Handicap_P8/src/fenetre/composants/AbstractJPanel.java - Eclipse". The main window displays the Java code for `AbstractJPanel.java`. The code defines a method `definirEtAjouterChamp` that adds components to a panel based on their type. It handles `JComponent`, `JTextComponent`, `JToggleButton`, `JTextField`, and `JTextArea` types, setting properties like bounds, visibility, and background colors. It also handles `JTabbedPane` by adding tabs. The code uses `if` statements to check component types and `else if` statements to handle specific cases like `JTabbedPane`.

```
80     private void definirEtAjouterChamp(JComponent composant, int x, int y, int width, int height, Color color, boolean visible, boolean enabled) {
81         composant.setBounds(x, y, width, height);
82         composant.setVisible(visible);
83         if(!(composant instanceof JDateChooser)) {
84             composant.setEnabled(enabled);
85         }
86         if(composant instanceof JTextComponent) {
87             ((JTextComponent) composant).setEditable(editable);
88             ((JTextComponent) composant).setText(chaine);
89             composant.setBackground(color);
90         }
91         if(composant instanceof JToggleButton) {
92             ((JToggleButton) composant). setSelected(selected);
93         }
94         add(composant);
95     }
96     if(composant instanceof JTextField) {
97         ((JTextField) composant).setColumns(10);
98         if(composant instanceof JTextFieldIdentifiant) {
99             composant.addMouseListener((MouseListener) event);
100        }
101    }
102    else if(composant instanceof JTextArea) {
103        ((JTextArea) composant).setLineWrap(true);
104        ((JTextArea) composant).setWrapStyleWord(true);
105        pane.setViewportView(composant);
106    }
107 }
```

This screenshot shows the continuation of the code from the previous one. It includes additional logic for handling `JTabbedPane` components, specifically adding tabs to it. It also includes code for `DureeHandicap` components, such as adding change listeners and setting selected states based on chain values.

```
103     }
104     else if(composant instanceof JTextArea) {
105         ((JTextArea) composant).setLineWrap(true);
106         ((JTextArea) composant).setWrapStyleWord(true);
107         pane.setViewportView(composant);
108     }
109     else if(composant instanceof JTabbedPane) {
110         for(int i=0;i<titresOnglets.size();i++) {
111             for(int j=0;j<onglets.size();j++) {
112                 if(i==j) {
113                     ((JTabbedPane) composant).addTab(titresOnglets.get(i), onglets.get(j));
114                 }
115             }
116         }
117     }
118     else if(composant instanceof DureeHandicap) {
119         ((DureeHandicap) composant).addChangeListener((ChangeListener) event);
120         if(!chaine.equals("")) {
121             if(chaine.equals("Non")) {
122                 ((DureeHandicap) composant). setSelected(false);
123             }
124             else {
125                 ((DureeHandicap) composant). setSelected(true);
126                 if(composant instanceof HandicapTemporaire) {
127                     handicapParticulier.setText(chaine);
128                 }
129             }
130         }
131     }
132 }
```

The screenshot shows the Eclipse IDE interface on a Mac OS X system. The title bar reads "Eclipse File Edit Source Refactor Navigate Search Project Run Window Help 69 % Dim. 28 août 11 15 54". The active window is titled "Identite.java" and contains the following Java code:

```
199
200     public Identite() throws LongueurDifférenteListesException, NullArgumentException {
201         this.nouveau=true;
202
203         JScrollPane scrollPane = new JScrollPane(this, JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_AS_
204         scrollPane.setBounds(this.getBounds());
205
206         setLayout(null);
207
208         JLabel lblNewLabel = new JLabel("Date de création du dossier :");
209
210         gestionChampsEtExceptions(lblNewLabel, 10, 11, 190, 14, null, true, true, null, null, null, null, null, null, null, null,
211         textField = new JDateChooser();
212
213         gestionChampsEtExceptions(textField, 204, 8, 150, 20, null, true, true, null, null, null, null, null, "", null, null, null
214         JLabel lblNewLabel_1 = new JLabel("Dernière mise à jour :");
215
216         gestionChampsEtExceptions(lblNewLabel_1, 10, 36, 150, 14, null, true, true, null, null, null, null, null, null, null, null,
217         textField_1 = new JDateChooser();
218
219         gestionChampsEtExceptions(textField_1, 204, 39, 150, 20, null, true, true, null, null, null, null, null, "", null, null, nu
220         JLabel lblNewLabel_2 = new JLabel("N° Etudiant :");
221
222         gestionChampsEtExceptions(lblNewLabel_2, 10, 73, 90, 14, null, true, true, null, null, null, null, null, null, null, null,
```

The code implements a constructor for a class named "Identite" that initializes a scroll pane and several labels and date pickers. It uses a "gestionChampsEtExceptions" method to handle input validation for each field.

X



**Documentation technique du
projet de gestion des
informations des étudiants en
situation de handicap**

1 En quoi consiste le projet ?

Il s'agissait, dans le cadre d'un stage de 3 mois (01/06/2016-31/08/2016), de commencer à développer une application de gestion des informations des étudiants en situation de handicap de l'université Paris 8 Vincennes – Saint-Denis.

2 Dans quel but ce projet a-t-il été lancé ?

Le personnel de l'Accueil Handicap de l'université réalise et/ou met à jour, jusqu'à présent, chaque dossier en format papier. Cela a pour inconvénients principaux :
-une perte de temps lorsqu'il s'agit de retrouver le dossier d'un étudiant donné, bien que ceux-ci soient bien classés et rangés ;
-une mise à jour qui nécessite parfois d'ajouter de nouvelles feuilles au dossier ou d'en archiver, ce qui peut induire des pertes, le nombre d'étudiants en situation de handicap étant assez conséquent (300 étudiants environ, tous campus compris).

Ces informations sont ensuite répertoriées et synthétisées dans un très grand tableau Excel comportant un certain nombre de colonnes correspondant aux informations de l'étudiant (type de handicap, aménagements des cours, des examens et/ou de la formation, avantages et prestations accordés par la MDPH, etc...).

Ce projet a pour but principal de permettre la création et la mise à jour facilitées des dossiers des étudiants handicapés de l'université via un logiciel qui se veut le plus intuitif et le plus ergonomique possibles.

NB : La fonctionnalité de suppression n'est pas prise en charge par le logiciel car tous les dossiers des étudiants, handicapés ou pas, sont conservés à vie au cas où ceux-ci reprendraient des études au cours de leur vie.

3 Première approche et contraintes

3.1 Une première approche bien différente de celle d'aujourd'hui...

Le projet aurait dû, idéalement, être développé au moyen de la plate-forme AMP (Apache, MySQL, PHP) en fonction de l'OS (Operating System ou Système d'Exploitation en français) : WAMP pour Windows, MAMP pour Mac ou XAMP pour Linux.

Apache est un programme permettant de transformer son ordinateur en serveur afin de pouvoir exécuter localement des applications web écrites au moyen du langage PHP.

MySQL est un SGBD (Système de Gestion de Base de Données) qui permet de créer des bases de données, de les remplir au moyen d'un schéma relationnel défini au préalable afin d'assurer l'intégrité des données, de les interroger et/ou de les modifier (modifier ou supprimer des données) au moyen de requêtes écrites en langage SQL (Structured Query Language ou Langage de Requêtes Structurées).

PHP (Personnal Home Page) est un langage de programmation permettant de développer des applications web dynamiques (dont le contenu change en fonction des données renvoyées par les requêtes SQL).

L'idée aurait été de créer une base de données dont chaque table aurait contenu les informations de chaque écran de l'application, chaque information étant associée à un numéro d'étudiant clé étrangère dans toutes les tables sauf dans la table étudiant représentant la fiche d'identité où celui-ci aurait été une clé primaire. Cet identifiant est en effet unique et permet de retrouver n'importe quel étudiant.

Ceci aurait permis ensuite de récupérer les informations déjà disponibles sur la base de données Apogée afin de pouvoir les utiliser et fusionner notre base avec celle-ci.

Cette application aurait également pu être développée au moyen de Java Enterprise Edition (communément appelé J2E). En effet, l'application Apogée est elle-même développée en java. Cette solution aurait permis, tout comme le PHP de créer cette application sous forme d'un site web en utilisant une base de données et en permettant la création de pages dynamiques.

3.2 ...Causée par un certain nombre de contraintes techniques imposées.

Toutefois, ceci n'a pas pu se passer comme prévu car le responsable de la DSI ne veut pas que nous ayons accès à la base de données afin d'éviter les conflits et les erreurs que nous pourrions causer. De plus, les données sont très personnelles.

De ce fait, il a fallu développer l'application en travaillant sur des fichiers txt pour les listes déroulantes de certains écrans et csv pour les informations concernant les étudiants. Les données sur lesquelles nous travaillons pour le développement de l'application sont fictives.

Par ailleurs, l'application Apogée utilisant la base de données à laquelle l'accès nous est interdit est écrite en langage JAVA. De fait, le développement de notre application a été réalisée avec ce même langage.

4 Architecture du projet

Le projet est constitué, actuellement, de quatre packages :

-un package data contenant lui-même deux packages :

-un package txt contenant tous les fichiers txt servant à remplir les listes déroulantes de certains écrans ou les indications dans certains formulaires de l'application :

-le fichier amenagements.txt contient la liste de tous les aménagements humains (secrétaire, interprète en LSF, etc...) ou techniques (clés d'accès aux ascenseurs, aux toilettes pour les gens en fauteuil roulant par exemple) disponibles pour les personnes en situation de handicap, que ce soit pour les cours, les examens ou la formation. Il sert à indiquer les champs du formulaire de l'écran concernant les aménagements ;

-le fichier handicap.txt qui contient tous les types de handicaps répertoriés par l'accueil handicap. Il sert à indiquer les champs du formulaire de l'écran concernant le handicap ;

-le fichier materiel_personnel.txt regroupe tous les moyens matériels que l'étudiant peut posséder personnellement pour ses études et dont il se sert. Il sert à compléter une liste déroulante dans l'écran concernant ses aménagements ;

-le fichier materiel_service.txt regroupe tous les moyens matériels mis à la disposition des étudiants handicapés par le service Accueil Handicap de l'université. Il sert à remplir une liste déroulante dans l'écran concernant ses aménagements ;

-le fichier medecins.txt contient tous les noms des médecins travaillant au service de la médecine préventive de l'université. Il permet de remplir une liste déroulante se trouvant dans l'écran concernant les aménagements lorsqu'il s'agit de choisir le nom du médecin rencontré lors de la dernière visite à la médecine préventive afin de bénéficier des aménagements adéquats ;

-le fichier statut.txt contient les différents statuts possibles de l'avancement d'un dossier déposé à la MDPH (Maison Départementale des Personnes Handicapées) du département où réside l'étudiant. Il est à noter que NC signifie que le handicap de l'étudiant n'est pas renseigné et ne peut donc pas donner lieu à des informations concernant le dossier MDPH. Par ailleurs, le statut « Non » peut tout aussi bien révéler que le dossier a été rejeté, qu'il n'y a pas eu de demande, ou encore que le dossier est en cours ;

-le fichier taux_incapacite_mdpf contient les différents taux possibles accordés par la CDAPH (Commission des Droits et de l'Autonomie des Personnes Handicapées) suite au dépôt du dossier par l'étudiant à la MDPH. Celui-ci détermine les avantages et prestations auxquels a droit l'étudiant qui en fait la demande. Ce taux est déterminé lors de la réunion de la commission et est questionné en fonction des demandes de l'étudiant. Ainsi, la demande d'une RQTH (Reconnaissance de la Qualité de Travailleur Handicapé) peut être acceptée quel que soit le taux d'incapacité. Certaines aides telles que la carte européenne de stationnement ne sont pas attribuées en fonction du taux d'incapacité mais en fonction de la capacité qu'a la personne à se déplacer. Ainsi, on peut voir cette carte refusée à une personne dont le taux d'incapacité est de 80 % ou plus et accordée à une personne avec un taux d'incapacité inférieur. Voici maintenant les différents taux d'incapacité et leurs significations respectives :

-le taux d'incapacité inférieur à 50 % : il n'ouvre éventuellement l'accès qu'à une RQTH si celle-ci est justifiée car ce taux signifie une gêne occasionnelle et entraînant relativement peu de difficultés dans la vie courante ;

-le taux d'incapacité compris entre 50 et 79 % inclus : là, la gêne est considérée comme effectivement importante. La personne ayant ce taux doit recourir à certains aménagements spécifiques pour pouvoir réussir et s'intégrer. Les capacités pour les activités primaires de la vie telles que manger, se déplacer, se comporter de manière sensée sont conservées. Il peut ouvrir le droit à la carte de priorité, donnant un accès prioritaire aux places assises dans les transports et dans les files d'attente lorsque la CDAPH a reconnu à la personne une station debout pénible. Il ouvre également l'accès à l'AAH (Allocation aux Adultes Handicapés) si la CDAPH reconnaît à la personne ayant déposé le dossier une restriction substantielle et durable d'accès à un emploi compte tenu des effets du handicap. Enfin, l'accès à la PCH (Prestation Compensation au Handicap), qui est une aide financière permettant d'obtenir de l'argent afin de pouvoir payer certaines aides techniques telles que des logiciels adaptés (ZoomText, Jaws (synthèse vocale), etc...) ou humaines (aide au ménage dans le cadre de l'ATPD (Aide au Travail Personnel à Domicile)) si la MDPH reconnaît une activité de la vie courante dont la difficulté est absolue (la personne ne peut pas réaliser cette activité seule) ou deux activités dont la difficulté est grave (la personne y arrive mais très difficilement) ;

-le taux d'incapacité supérieur ou égal à 80 % : la gêne est considérée comme entraînant une grande gêne dans la vie courante concernant l'autonomie de la personne. Ce taux est attribué à toute personne ayant une déficience sensorielle totale ou sévère ou encore aux personnes ayant des difficultés dans les tâches primaires de la vie quotidienne (manger, se repérer dans le temps et dans l'espace, s'habiller, etc...). Elle ouvre accès de droit à la carte d'invalidité faisant office de carte de priorité et donnant, de fait, le droit à l'AAH et à la PCH. La carte d'invalidité donne droit à des réductions sur les parts d'impôts, à la gratuité de beaucoup de musées, à l'accès prioritaire aux logements sociaux, à des réductions sur les billets de train et à l'assistance voyageur handicapé dans les gares (Accès plus), les aéroports et les bateaux. Il est à noter qu'elle donne également accès à des sociétés de transporteurs qui conduisent la personne d'un point A à un point B.

Il est également important de noter que ces taux sont théoriques et qu'en pratique, ce n'est pas toujours appliqué. Il n'est donc pas rare de voir, par exemple, des personnes reconnues entre 50 et 79 % qui devraient être à 80 %, et cela même après un recours, tout dépendant du département où l'on se trouve. Enfin, il faut savoir, concernant l'AAH et la PCH, que celles-ci sont attribuées par la CAF, suite à la notification de la MDPH. Ces prestations peuvent être données en totalité, partiellement ou pas du tout, en fonction des revenus de la personne qui en fait la demande ;

-le fichier departements.txt contenant tous les départements français, DOM compris afin de sélectionner le département de résidence de l'étudiant ou de ses parents, car c'est de celui-là que dépend l'endroit où doit être déposé le dossier MDPH (en général, la ville où se trouve la préfecture) ;

-un package csv contenant tous les fichiers csv de données des étudiants en fonction de l'écran dans lequel on se trouve (un fichier pour chaque écran) :

-la première ligne de chaque fichier représente un libellé de champ de formulaire ;

-les champs sont soit renseigné par oui, non ou un texte dans certains écrans (projets par exemple) ou pour certains cas spécifiques (un handicap moteur non répertorié par exemple).

Le package data contient, par ailleurs, les classes de lecture/écriture des fichiers csv et des fichiers txt ;

-un package interfaces contenant deux interfaces respectivement de gestion des informations des fichiers txt et csv qui seront implémentées dans les classes manipulant des fichiers ;

-un package exceptions contenant deux exceptions personnalisés, l'une lorsque la longueur de la liste des onglets et celle des titres des onglets ne sont pas les mêmes et l'autre lorsque l'une des deux listes est à null. Cela se produit dans la classe AbstractJPanel qui contient une méthode protected qui sera partagée par tous les JPanel de l'application ;

-un package fenêtre qui contient la classe permettant de définir les caractéristiques de la fenêtre ainsi que trois packages :

-un package img contenant une image de maison qui est utilisé pour revenir à l'écran d'authentification de l'application quand on est dans l'écran d'accueil avec les onglets ;

-un package écrans contenant tous les JPanel représentant les contenus de la fenêtre de l'authentification jusqu'aux différents onglets de l'application ;

-un package composants contenant toutes les classes étendant des composants afin de traiter des cas particuliers dans la méthode permettant de définir les champs des formulaires dans la classe AbstractJPanel ;

-enfin, le package par défaut contient la classe principale (Main) permettant de lancer l'application ;

-un dossier poubelle où se trouve les premiers développements de l'application dans une seule et unique classe afin de montrer la difficulté si on n'applique pas la méthode du pattern Singleton avec la fenêtre (faire en sorte qu'il n'y ait qu'une seule instance réutilisable d'un objet durant l'utilisation de l'application) ;

-un dossier rapport contenant le journal de bord de stage, le plan du rapport, les annexes, le rapport en lui-même ainsi que le logo de l'université pour la page de garde et les bugs non résolus à ce jour ;

-un dossier doc contenant toute la javadoc du projet (documentation java détaillée, sous forme de page web, des classes, attributs et méthodes) ;

-un dossier test contenant une classe de tests unitaires sur les données retournées en fonction d'un tableau passé en paramètre et l'index demandé ;

-un dossier com contenant la librairie JCalendar pour la gestion des champs de type date ; un dossier bin contenant tous les fichiers semi-compilés du projet (.class) afin qu'ils soient ensuite traduit en binaire et exécutable par l'ordinateur.

5 Choix de programmation

Dans un souci du respect des normes de la programmation orientée objet, l'idée était principalement de séparer tous les composants dans plusieurs fichiers afin de rendre le code le plus lisible et le plus facilement modifiable possibles pour les développeurs qui reprendraient le projet derrière.

Au début, après plusieurs tentatives infructueuses, les classes représentant chaque onglet ainsi que la classe d'authentification et d'accueil se trouvaient déclarées private dans la classe Fenetre. Même si cela fonctionnait, le code était très alourdi et devenait de moins en moins lisible et maintenable au fur et à mesure de l'avancement du code.

Ensuite, le pattern singleton s'est révélé très utile en permettant de créer une seule et unique instance de la classe Fenetre à l'intérieur même de la classe via une méthode déclarée static qui pouvait être appelée de n'importe où car déclarée public.

Cela s'est révélé bien pratique pour diviser le code comme envisagé au début. Ensuite, il s'agit, pour le moment, de réaliser un refactoring maximal afin d'éviter au maximum les répétitions de code à plusieurs endroits afin d'éviter des sources d'erreurs engendrées par du code que l'on modifie à un endroit et que l'on oublie de modifier à d'autres. Or, pour arriver à cela, il a fallu créer des classes étendant des composants graphiques notamment sans rien apporter de plus mais simplement pour traiter des cas particulier dans la grande méthode de gestion des composant dans les panels.

6 Ce qui fonctionne à l'heure actuelle

Les écrans de l'identité, du handicap, des aménagements, des projets, de la fiche ministérielle (MENESR) et du carnet de visite devraient être finis en lecture seule.

7 Ce qu'il reste à faire

Développer les écrans qui contiennent des tableaux en utilisant des JTable et des modèles qui sont associés concernant les données (ParcoursAnterieurP8, Inscription et NotesReussite).

Les fonctionnalités d'ajout et de modification devront aussi être mises en place, la seule fonctionnant correctement étant la lecture seule des données lorsqu'un étudiant est trouvé.

Il s'agira également de finir le développement de l'écran Menesr qui devra être modifié en déplaçant les informations sur le centre médico-social se trouvant dans ParcoursAnterieurP8 dans le fichier Menesr et en mettant une comboBox (liste déroulante) à la place de boutons radio pour ce cas précis.

Enfin, les barres de défilement doivent être résolues au plus vite car tout l'écran doit être visible pour pouvoir permettre une création, une mise à jour ou une consultation facilitées et testées globalement.

8 Axes d'amélioration

Faire en sorte que le champ N° Etudiant ne soit pas éditable si les champs nom et/ou prénom contiennent du texte saisi (pour l'instant, seul l'inverse est géré (les champs nom et prénom ne sont plus éditables une fois le numéro d'étudiant saisi)).

L'application pourrait être retravaillée et développée en J2E avec une base de données si, par la suite, le responsable de la DSI était d'accord suite à la présentation du début des développements actuels.

Annexe F : Morceaux de la javadoc du projet

AbstractJPanel

file:///Users/alexis/git/Gestion_Infos_Accueil_Handicap_P8/doc/index.html

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

fenetre.composants

Class AbstractJPanel

```
java.lang.Object
    java.awt.Component
        java.awt.Container
            javax.swing.JComponent
                javax.swing.JPanel
                    fenetre.composants.AbstractJPanel
```

All Implemented Interfaces:

```
java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible
```

Direct Known Subclasses:

```
Accueil, Amenagements, AmenagementsCours, AmenagementsExamens, Authentification,
CarnetDeVisiteEtCommentaires, Handicap, Identite, Inscription,
ListeEtudiantsAvecNomOuPrenomCommuns, Menesr, NotesReussite, ParcoursAnterieurP8, Projets
```

public abstract class **AbstractJPanel**
extends javax.swing.JPanel

Classe héritée par tous les JPanel de l'application.

AbstractJPanel

file:///Users/alexis/git/Gestion_Infos_Accueil_Handicap_P8/doc/index.html

Constructors

Constructor and Description

AbstractJPanel()

Method Summary

| All Methods | Instance Methods | Concrete Methods |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Modifier and Type | Method and Description | |
| private void | definirEtAjouterChamp (javax.swing.JComponent composant, int x, int y, int width, int height, java.awt.Color color, boolean visible, boolean enabled, java.lang.Boolean editable, java.lang.Boolean selected, java.util.ArrayList<java.lang.String> titresOnglets, java.util.ArrayList<AbstractJPanel> onglets, java.util.EventListener event, java.lang.String chaine, javax.swing.JTextField handicapParticulier, DroitFait statutAmenagement, javax.swing.JScrollPane pane) Méthode centrale permettant de placer les composants sur chaque JPanel implémentant cette classe | |
| protected void | gestionChampsEtExceptions (javax.swing.JComponent composant, int x, int y, int width, int height, java.awt.Color color, boolean visible, boolean enabled, java.lang.Boolean editable, java.lang.Boolean selected) | |

Overview

file:///Users/alexis/git/Gestion_Infos_Accueil_Handicap_P8/doc/index.html

Rechercher

All Classes

Packages

- <unnamed package>
- data
- exceptions
- fenetre
- fenetre.composants
- fenetre.ecrans

All Classes

- AbstractJPanel
- Accompagnement
- AccompagnementExterieurApres
- AccompagnementExterieurAvant
- Accueil
- AffectationPrioritaire
- AidesHumainesCours
- AidesHumainesExamens
- AidesTechniquesCours
- AidesTechniquesExamens
- Amenagements
- AmenagementsCours
- AmenagementsExamens
- AmenagementsFormation
- Audition
- Authentification
- Autisme
- ButtonGroupAmeliorer
- CarnetDeVisiteEtCommentaires
- DonneesTabbedPane
- Droit

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Packages

| Package | Description |
|--------------------|-------------|
| data | |
| exceptions | |
| fenetre | |
| fenetre.composants | |
| fenetre.ecrans | |
| interfaces | |

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Identite

file:///Users/alexis/git/Gestion_Infos_Accueil_Handicap_P8/doc/index.html

Rechercher

All Classes

Packages

- <unnamed package>
- data
- exceptions
- fenetre
- fenetre.composants
- fenetre.ecrans

- Fait
- FaitSpecifiqueHandicapMoteur
- FamilleHandicapNonDefinie
- Fenetre
- FormationParCorrespondance
- GestionCasesCochesAmenagement
- GestionFichierCSV
- GestionFichierTXT
- GrouperBoutons
- Handicap
- HandicapAPreciser
- HandicapParticulier
- HandicapTemporaire
- Identite
- Inscription
- JTextFieldIdentifiant
- LectureFichierCSV
- LectureFichierTest
- LectureFichierTXT
- ListeEtudiantsAvecNomOuPrenom
- LongueurDifférenteListesException
- Main
- Menesr

| | |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| private LectureFichierCSV | fichierIdentite |
| | Variable permettant de travailler avec le fichier contenant toutes les informations gérées par rapport à chaque étudiant |
| private boolean | nouveau |
| | Indique si l'étudiant est nouveau ou pas |
| private java.lang.String | numEtudiant |
| | Le numéro d'étudiant |
| private Sexe | rdbtnM |
| | Bouton Monsieur |
| private Sexe | rdbtnMme |
| | Bouton Madame |
| private com.toedter.calendar.JDateChooser textField | dateCréationDossier |
| | Date de création du dossier |
| private com.toedter.calendar.JDateChooser textField_1 | dateDernièreMiseAJourDossier |
| | Date de dernière mise à jour du dossier |
| private javax.swing.JTextField | textField_10 |
| | Téléphone |
| private javax.swing.JTextField | textField_11 |
| | E-mail |
| private javax.swing.JTextField | textField_12 |

Annexe G : Journal de bord du stage

Journal de bord stage M1 informatique (1^{er} juin 2016 – 31 août 2016)

Semaine 1 (1^{er} juin – 3 juin) :

- création des fichiers de données de l'application ;
- installation des outils nécessaires au développement de l'application ;
- mise en place d'un dépôt distant sur lequel se trouvent toutes les sources du projet.

Semaine 2 (6 juin – 10 juin)

- rattrapages de certaines matières (6 et 7 juin) ;
- soutenance de projet tuteuré (9 juin) ;
- développement de l'écran d'authentification fonctionnel.

Semaine 3 (13 juin – 17 juin) :

- premier développement des écrans identité, projets, handicap, aménagements et carnet de visites ;
- développement de l'écran d'accueil avec des onglets représentant chaque écran de l'application ;
- premier point avec référent de stage pour discuter des difficultés et avoir des conseils (15 juin) ;
- première réunion avec l'équipe pour discuter des développements pour les améliorer par la suite (17 juin)

Semaine 4 (20 juin – 24 juin) :

- passage d'un partiel en fin de journée (journée essentiellement de révisions le 20 juin) ;
- ajout d'un bouton « Accueil » en forme de maison pour revenir à la première fenêtre de l'application ;
- champs de formulaire d'identité qui ne changent pas grisés ;
- fichiers de données initiaux complétés

Semaine 5 (27 juin – 1^{er} juillet) :

- pot de départ d'un des membres de l'équipe (27 juillet) ;
- création de nouveaux fichiers de données pour gérer les écrans où il y a des tableaux ;
- recherche d'informations sur le site de Paris 8 permettant d'apporter des informations au rapport de stage

Semaine 6 (4 juillet – 8 juillet) :

- création d'un bouton « Nouveau » dans la page d'authentification pour ajouter un étudiant + gestion du formulaire (grisé ou non) en fonction de l'étudiant s'il existe ou si on le crée ;
- tentatives de mise en place des barres de défilement quand c'est nécessaire ;

-point avec référent de stage pour discuter des difficultés et trouver des solutions (6 juillet) ;
-première ébauche de plan pour le rapport de stage

Semaine 7 (11 juillet – 15 juillet) :

- modification de l'écran d'aménagements en fonction des besoins de l'équipe ;
- point avec l'équipe pour discuter de l'avancement du projet et des modifications à apporter (13 juillet matin) ;
- discussion des difficultés et tentatives de résolution avec le référent de stage (13 juillet après-midi) ;
- pont le 15 juillet

Semaine 8 (18 juillet – 22 juillet) :

- modification de l'écran d'aménagements : séparation des aménagements en deux onglets, un pour les cours, l'autre pour les examens ;
- développement de l'écran MENESR pour l'enquête ministérielle sous forme de cases à cocher et non de tableau ;
- point avec l'équipe pour discuter de l'avancement du projet (20 juillet) annulé car référent malade ;
- début du refactoring complet du code afin de diviser le code dans le but de le rendre plus lisible et plus facilement modifiable et corigeable ;
- début du développement des écrans « parcours antérieur à Paris 8 » et inscriptions + lecture de tutoriels sur l'utilisation de l'implémentation des tableaux graphiques en java (JTable) afin de les utiliser dans ces écrans et dans celui concernant les notes de l'année en cours.

Semaine 9 (25 juillet – 29 juillet) :

- séparation des écrans grâce au pattern singleton permettant de ne créer qu'une seule instance de la fenêtre et de pouvoir y accéder et modifier son affichage (écran vu par l'utilisateur) ;
- création de fichiers txt permettant, pour les prochains développeurs, de les utiliser pour charger les champs de certains formulaires dynamiquement et remplir les listes déroulantes.

Semaines 10, 11 et 12 : université fermée, donc travail effectué à la maison.

Semaine 10 (1^{er} août – 5 août) :

- refactoring de la création des écrans en créant une classe abstraite permettant de positionner les composants et leur donner des comportements grâce à une seule méthode appelée par les classes filles

Semaine 11 (8 août – 12 août) :

- modification des écrans en utilisant la méthode développée dans la classe abstraite

Semaine 12 (15 août – 19 août) :

- continuation du refactoring amenant parfois à modifier la méthode au cas où il y aurait un paramètre à rajouter (ce qui a été le cas plusieurs fois). Du coup, modification de toutes les classes utilisant cette méthode ;

- rédition de la documentation technique du projet ;
- modification du plan de rapport ;
- division du code au sein d'une grosse méthode en plusieurs morceaux afin d'éviter les répétitions (méthode Suivant dans la classe Authentification)

Semaine 13 (22 août – 26 août) :

- continuation et fin du refactoring des classes et méthodes même si des choses sont encore à améliorer ou modifier (écrans Menesr et ParcoursAnterieurP8), voire compléter (listes déroulantes de l'écran Menesr) ;
- rédition et génération de la javadoc du projet ;
- rédition du rapport

Semaine 14 (29 août – 2 septembre) :

- continuation de la rédition + mise en page + lecture ;
- rapport rendu (1^{er} septembre)

Semaine 15 (5 septembre – 9 septembre) :

- soutenance de rapport (5 septembre)

Annexe H : Bibliographie/sitographie/filmographie

Comment ça marche : <http://www.commentcamarche.net/>

OpenClassrooms :

https://openclassrooms.com/dashboard?utm_source=google&utm_medium=cpc&utm_term=search&utm_campaign=308022735

Club des développeurs : <http://www.developpez.com/>

Support Microsoft Office : <https://support.office.com/fr-fr>

Site de l'université : <https://www.univ-paris8.fr/>

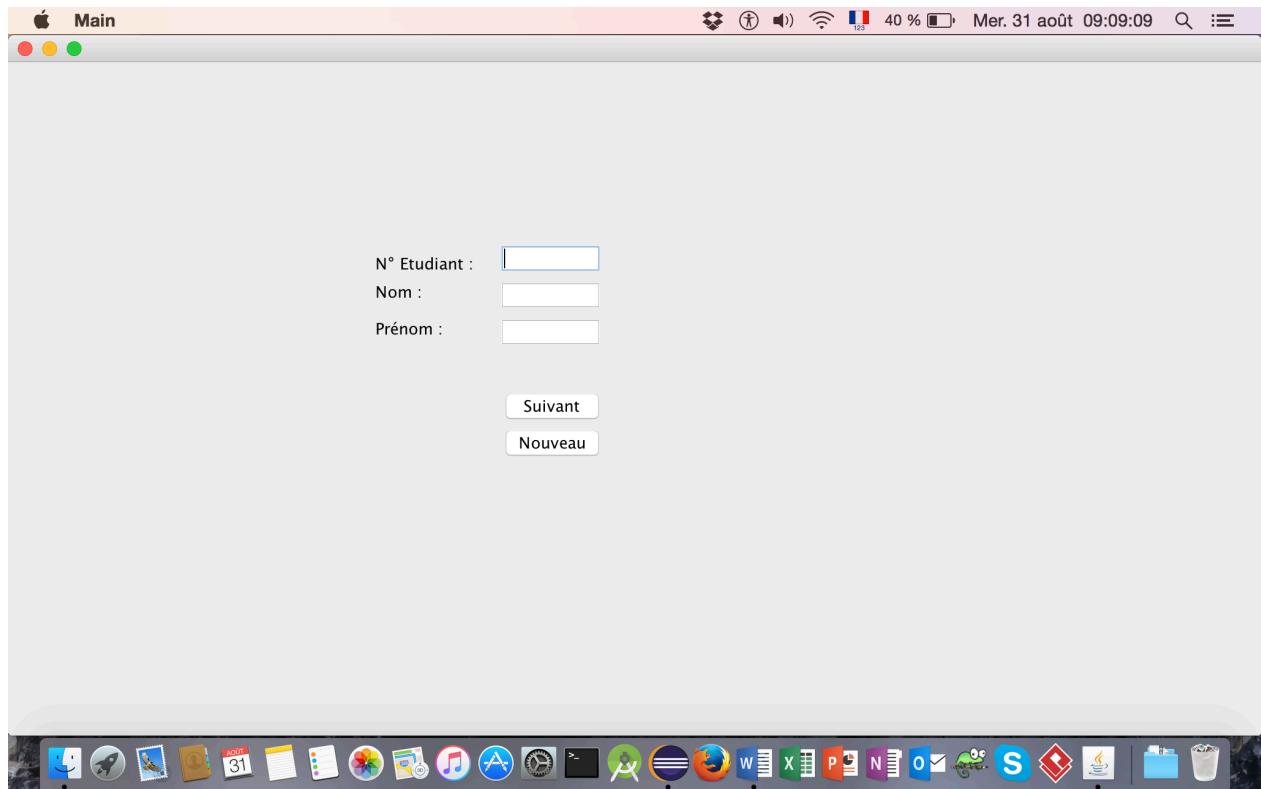
Documentaire « Vincennes, l'université perdue » : <http://www.documentaires-streaming.com/vincennes-luniversite-perdue/>

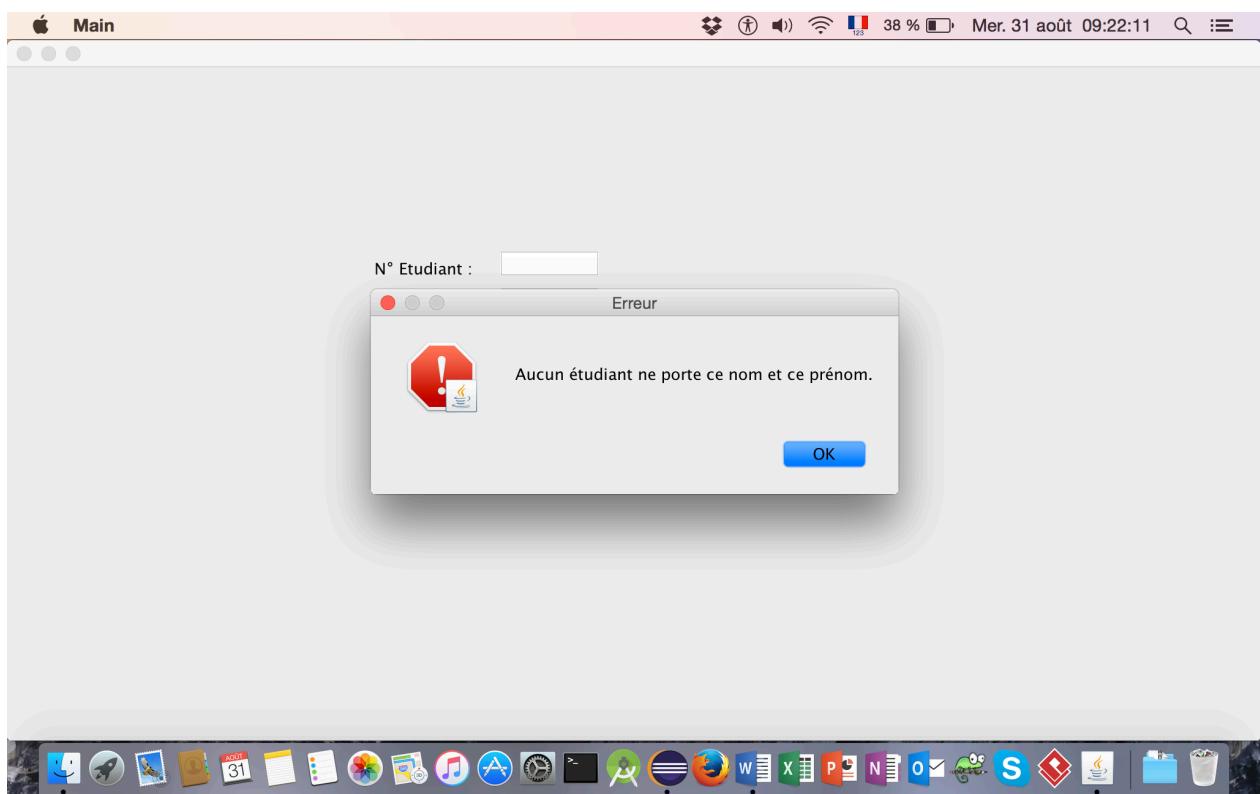
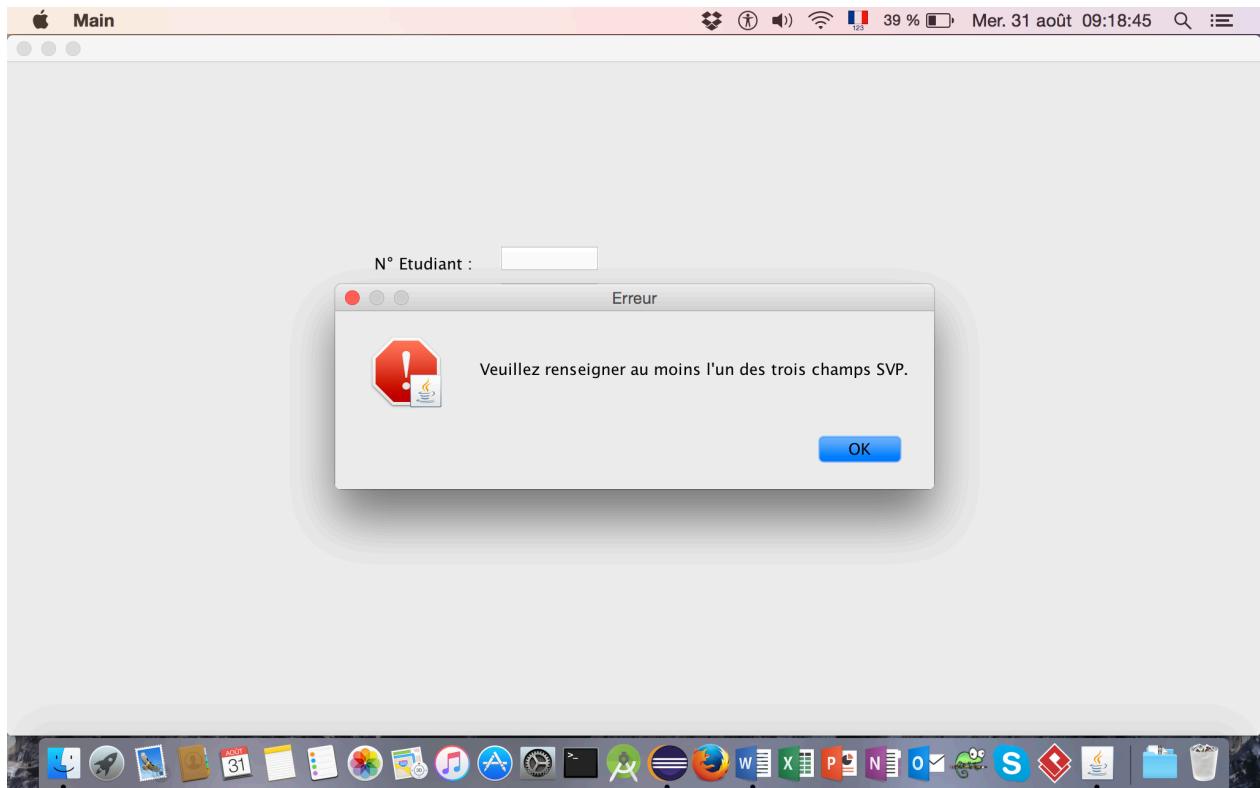
Annexe I : Bugs non résolus à ce jour

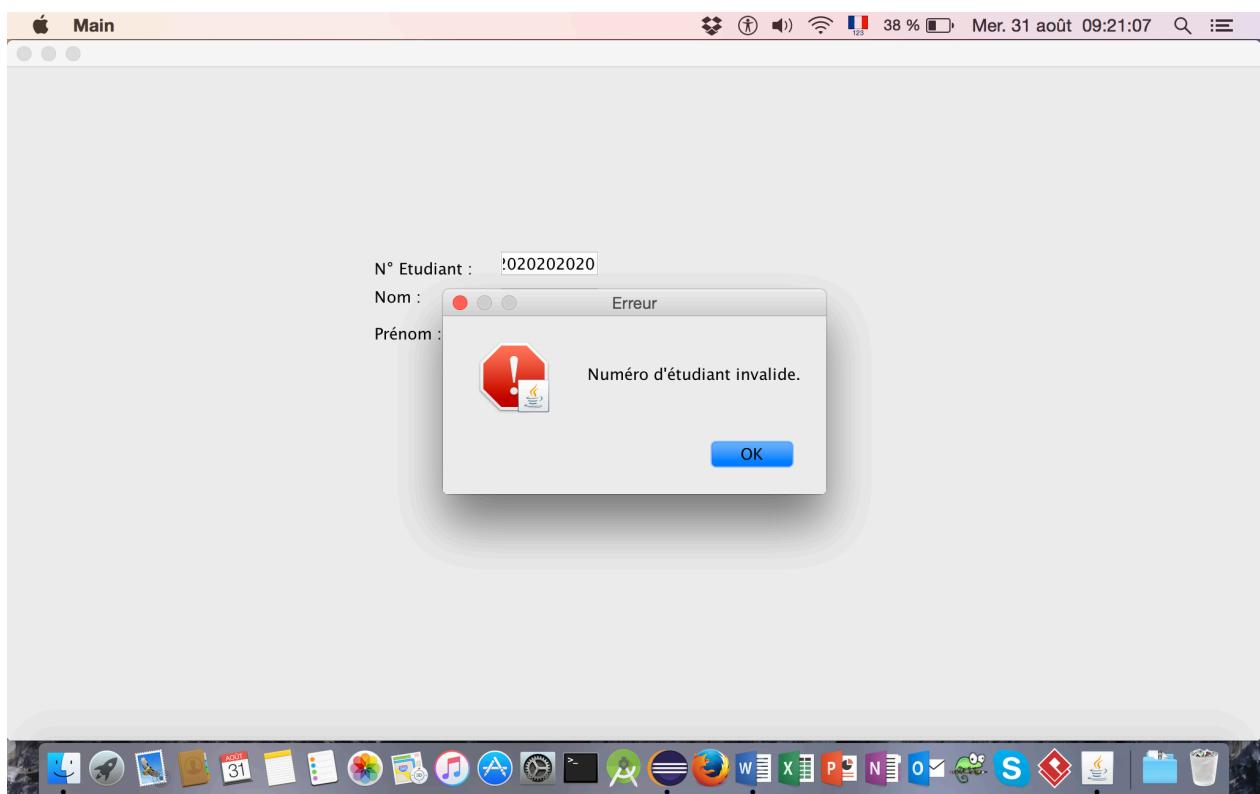
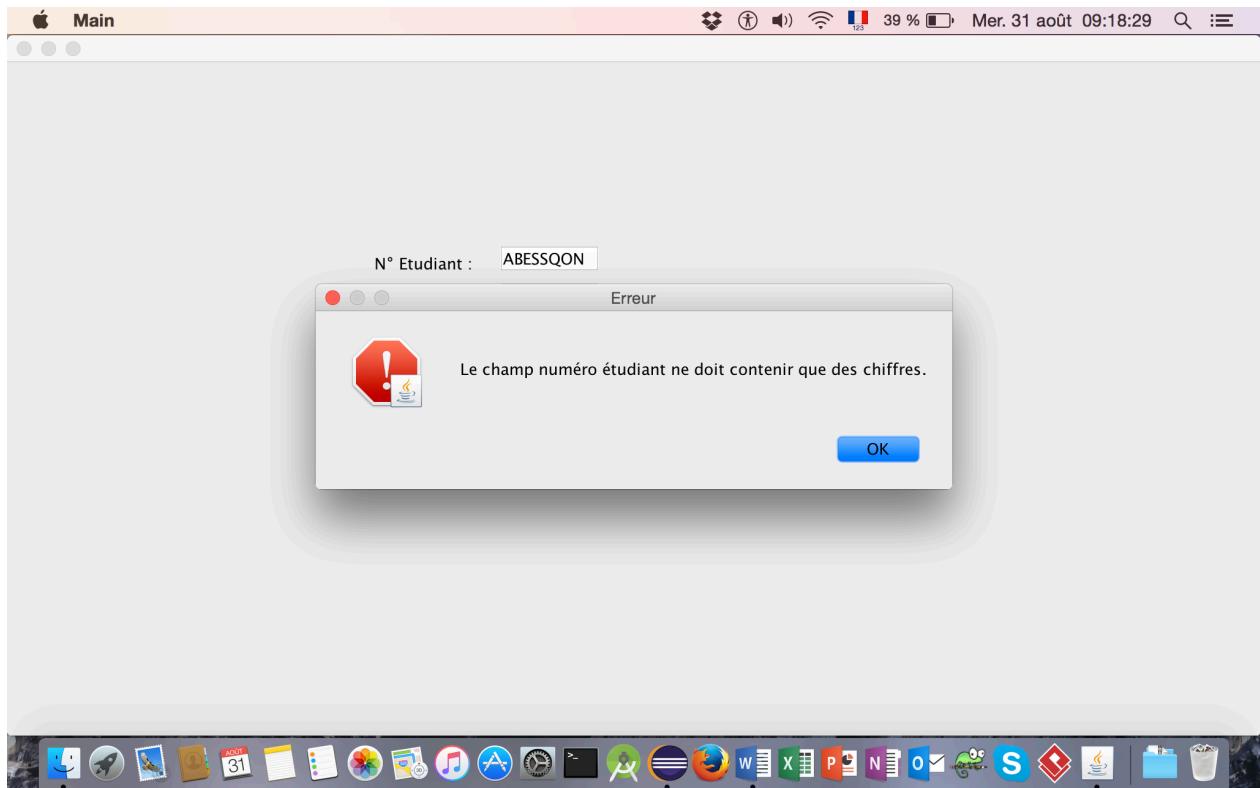
Liste des bugs non résolus à ce jour dans l'application de gestion d'informations des personnes en situation de handicap de l'université Paris 8 :

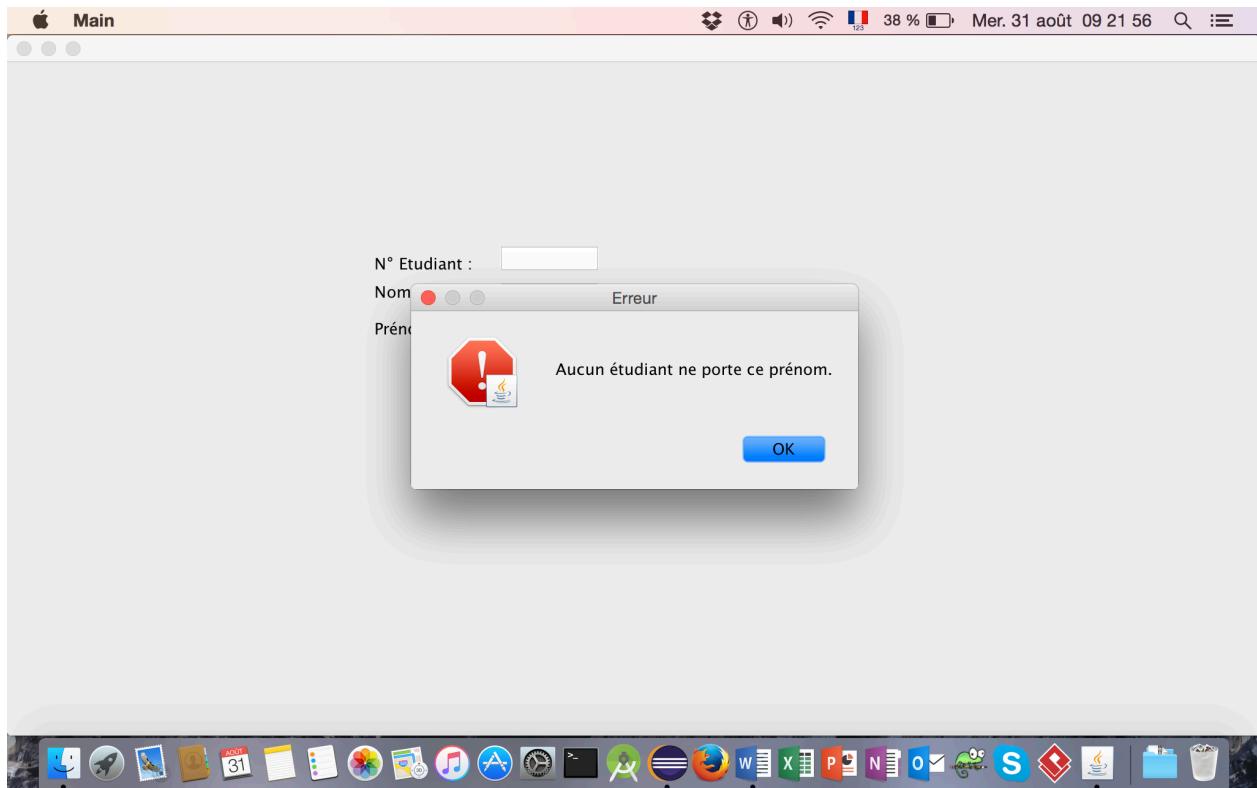
- quand on entre un nom et un prénom et que l'une des deux informations est commune à plusieurs personnes dans le fichier, l'écran d'accueil ne s'affiche pas mais la liste des personnes qui ont des caractéristiques en commun ;
- les barres de défilement n'apparaissent pas sur les fenêtres de l'application, ce qui ne permet pas d'accéder à la totalité de l'écran ;
- dans l'onglet examens de l'écran aménagements, le champ de date n'apparaît pas concernant la clé des WC accessibles aux personnes handicapées moteurs lorsque la case Fait est cochée dans la consultation (alors qu'elle apparaît bien dans le cadre de la création ou de la mise à jour).

Annexe J : Quelques captures d'écrans des écrans de l'application









A screenshot of a student profile creation form. The title bar says "Main". The toolbar includes icons for file operations, search, and others. The main window has tabs: Identité, Parcours, Inscription, Projets, Handicap, Aménagements, Menesr, Notes, and Autres. The "Identité" tab is selected. The form fields include:

- Date de création du dossier : [input field] Coordonnées d'une personne à contacter en cas d'urgence : [input field]
- Dernière mise à jour : [input field] Nom : [input field]
- N° Etudiant : [input field] Prénom : [input field]
- (radio buttons) M Mme
- Nom de naissance : [input field] Téléphone : [input field]
- Nom d'époux(se) : [input field] E-mail : [input field]
- Prénom : [input field] Fonc... : [input field]
- Date de naissance : [input field] Nom : [input field]
- Domicile : [input field] Prénom : [input field]
- Code postal : [input field] Télém... : [input field]
- Ville : [input field] E-mail : [input field]
- Téléphone : [input field] Commentaires : [text area]
- E-mail : [input field]

The Dock at the bottom contains various application icons.

Main

Ok Annuler

Identité Parcours Inscription Projets Handicap Aménagements Menesr Notes Autres

Date de création du dossier : 7 juil. 2010 Coordonnées d'une personne à contacter en cas d'urgence :

Dernière mise à jour : 1 juin 2016 Nom : DUPONT

N° Etudiant : 11020304 Prénom : Bernadette

M Mme

Nom de naissance : DUPONT Téléphone : 33660708090

Nom d'époux(se) : DUBOIS-DUPONT E-mail : te.dupont@yahoo.com

Prénom : Jean Fonction : Médecin traitant

Date de naissance : 1 janv. 1992 Nom : LAPEYRE

Domicile : 12 rue du savoir Prénom : Gérard

Code postal : 93200 Téléphone : 33158102030

Ville : Saint-Denis E-mail : gerard.lapeyre@orange.fr

Téléphone : 33123456789 Commentaires : Personne diabétique

E-mail : jean.dupont@yahoo.com



Main

Ok Annuler

Identité Parcours Inscription Projets Handicap Aménagements Menesr Notes Autres

Date de la visite à la Médecine Préventive

Nom du médecin : Franck LACROIX

Cours Examens

Aides humaines

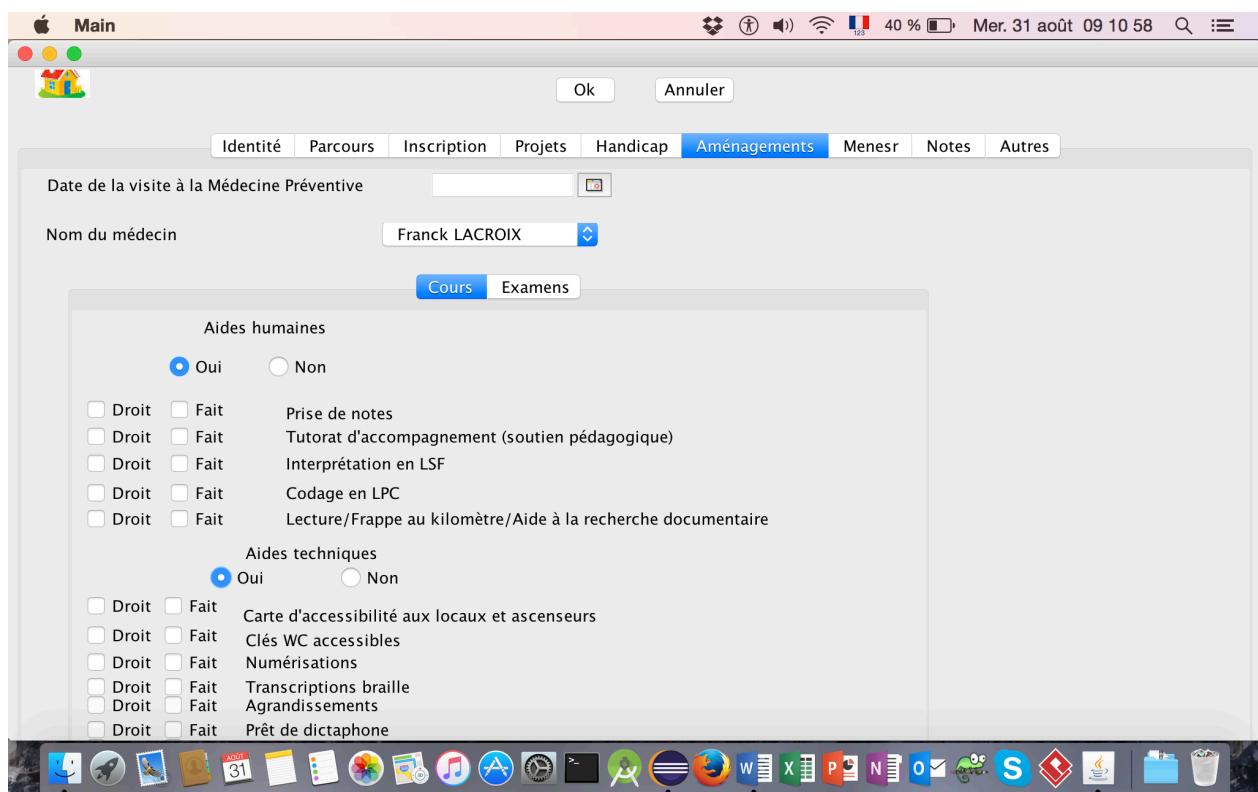
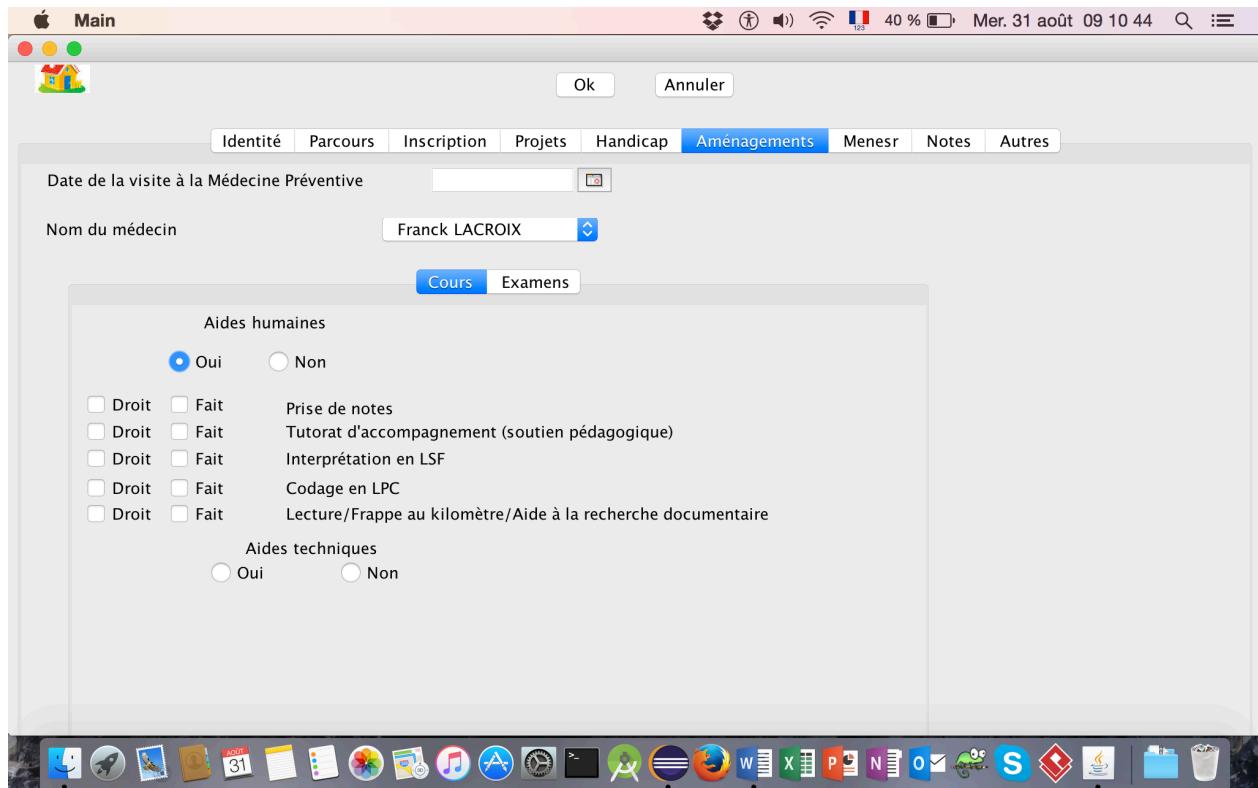
Oui Non

Aides techniques

Oui Non



XXX



Main

Ok Annuler

Identité Parcours Inscription Projets Handicap Aménagements Menesr Notes Autres

Date de la visite à la Médecine Préventive

Nom du médecin Franck LACROIX

Cours Examens

Aides humaines

Oui Non

| | | |
|--------------------------------|-------------------------------|--------------------------------------------------------------|
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Prise de notes |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Tutorat d'accompagnement (soutien pédagogique) |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Interprétation en LSF |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Codage en LPC |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Lecture/Frappe au kilomètre/Aide à la recherche documentaire |

Aides techniques

Oui Non

| | | | | |
|-------------------------------------------|------------------------------------------|----------------------------------------|-------------|----------------------|
| <input checked="" type="checkbox"/> Droit | <input checked="" type="checkbox"/> Fait | Carte d'accès aux locaux et ascenseurs | Remise le : | <input type="text"/> |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Clés WC accessibles | | |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Numérisations | | |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Transcriptions braille | | |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Agrandissements | | |
| <input type="checkbox"/> Droit | <input type="checkbox"/> Fait | Prêt de dictaphone | | |

Main

Ok Annuler

Identité Parcours Inscription Projets Handicap Aménagements Menesr Notes Autres

Non renseigné Handicap définitif

Handicap temporaire

Troubles moteurs :

Troubles visuels :

Troubles auditifs :

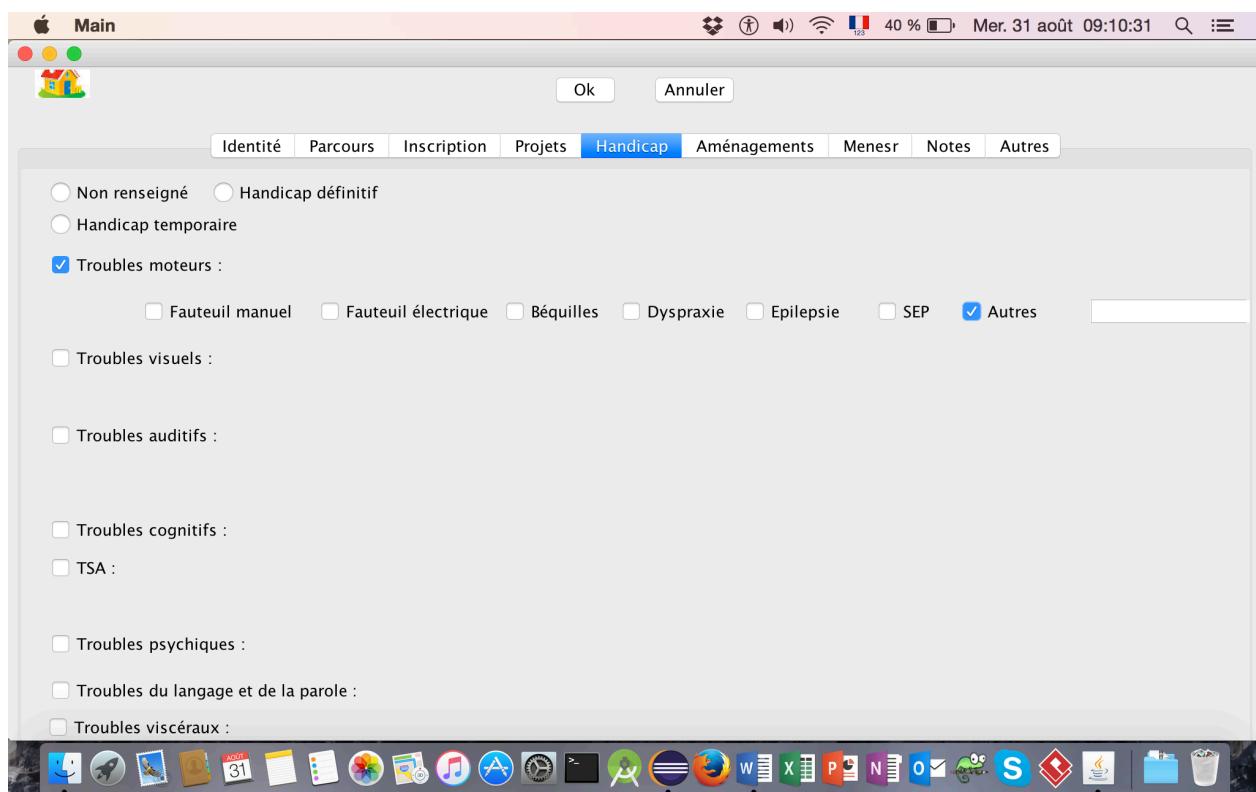
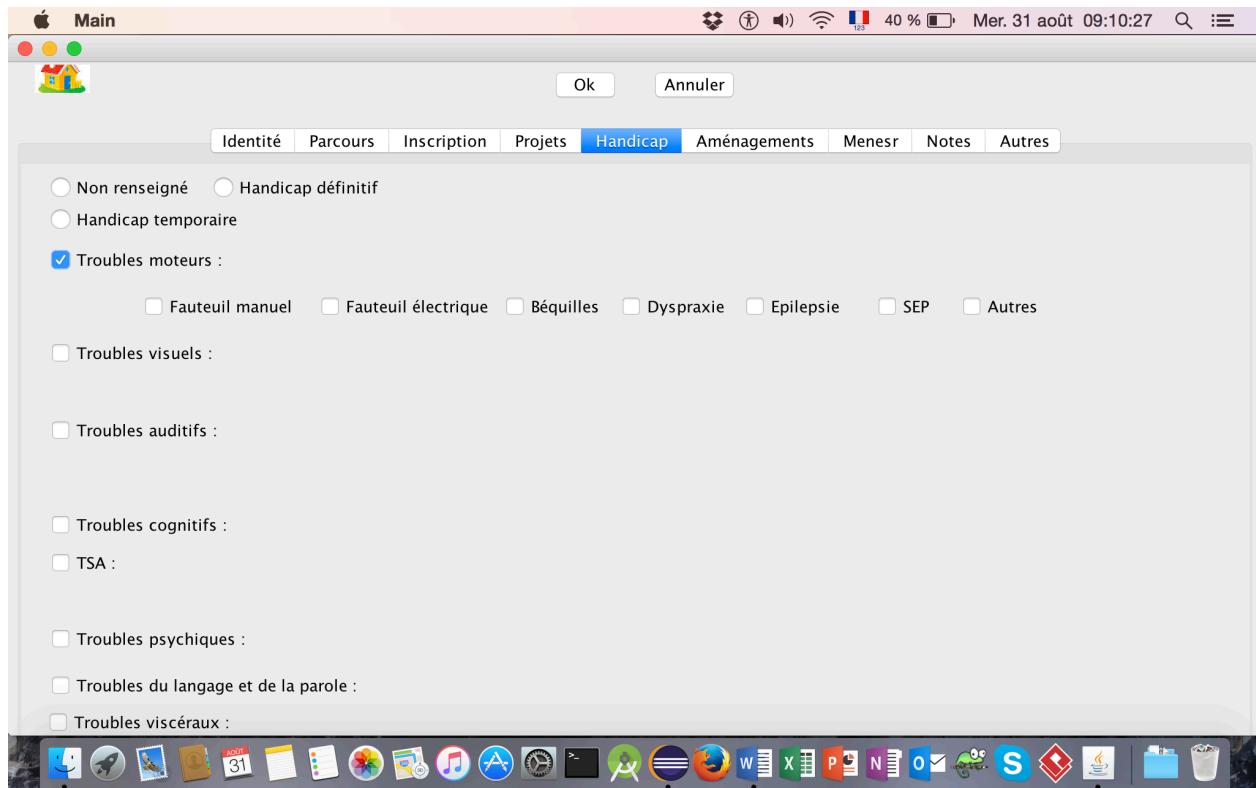
Troubles cognitifs :

TSA :

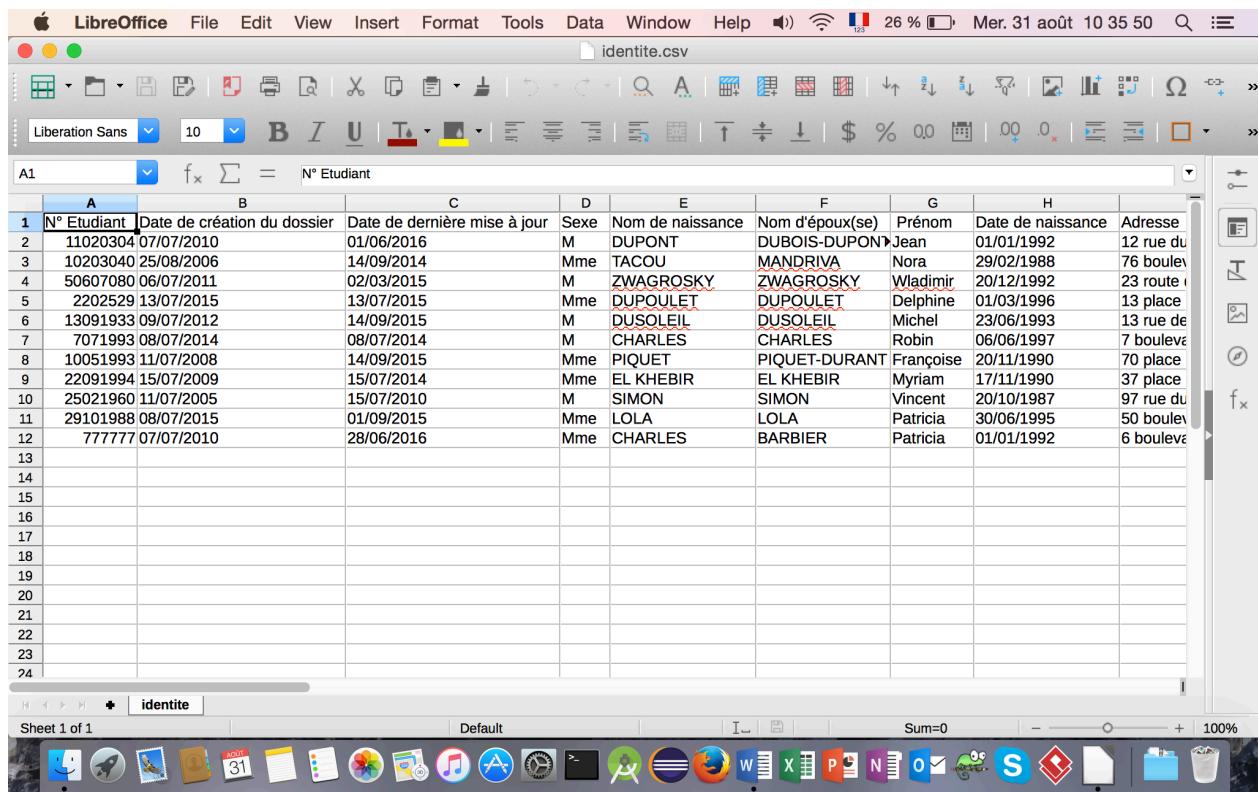
Troubles psychiques :

Troubles du langage et de la parole :

Troubles viscéraux :



Annexe K : Captures d'écrans de deux fichiers de données (ici, identité et projets) utilisés pour le développement de l'application



The screenshot shows a LibreOffice Calc spreadsheet titled "identite.csv". The data consists of 12 rows of student information, starting with a header row. The columns are labeled A through H. The data includes student ID, creation date, last update date, gender, birth name, spouse name, first name, birth date, and address.

| | A | B | C | D | E | F | G | H | |
|----|-------------|-----------------------------|------------------------------|------|------------------|-----------------|-----------|-------------------|--------------|
| 1 | N° Etudiant | Date de création du dossier | Date de dernière mise à jour | Sexe | Nom de naissance | Nom d'époux(se) | Prénom | Date de naissance | Adresse |
| 2 | 11020304 | 07/07/2010 | 01/06/2016 | M | DUPONT | DUBOIS-DUPON | Jean | 01/01/1992 | 12 rue du |
| 3 | 10203040 | 25/08/2006 | 14/09/2014 | Mme | TACOU | MANDRIVA | Nora | 29/02/1988 | 76 boulevard |
| 4 | 50607080 | 06/07/2011 | 02/03/2015 | M | ZWAGROSKY | ZWAGROSKY | Wladimir | 20/12/1992 | 23 route |
| 5 | 2202529 | 13/07/2015 | 13/07/2015 | Mme | DUPOULET | DUPOULET | Delphine | 01/03/1996 | 13 place |
| 6 | 13091933 | 09/07/2012 | 14/09/2015 | M | DUSOLEIL | DUSOLEIL | Michel | 23/06/1993 | 13 rue de |
| 7 | 7071993 | 08/07/2014 | 08/07/2014 | M | CHARLES | CHARLES | Robin | 06/06/1997 | 7 boulevard |
| 8 | 10051993 | 11/07/2008 | 14/09/2015 | Mme | PIQUET | PIQUET-DURANT | Françoise | 20/11/1990 | 70 place |
| 9 | 22091994 | 15/07/2009 | 15/07/2014 | Mme | EL KHEBIR | EL KHEBIR | Myriam | 17/11/1990 | 37 place |
| 10 | 25021960 | 11/07/2005 | 15/07/2010 | M | SIMON | SIMON | Vincent | 20/10/1987 | 97 rue du |
| 11 | 29101988 | 08/07/2015 | 01/09/2015 | Mme | LOLA | LOLA | Patricia | 30/06/1995 | 50 boulevard |
| 12 | 777777 | 07/07/2010 | 28/06/2016 | Mme | CHARLES | BARBIER | Patricia | 01/01/1992 | 6 boulevard |
| 13 | | | | | | | | | |
| 14 | | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | | | | | | | | | |
| 17 | | | | | | | | | |
| 18 | | | | | | | | | |
| 19 | | | | | | | | | |
| 20 | | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | | | | | | | | | |
| 23 | | | | | | | | | |
| 24 | | | | | | | | | |

LibreOffice File Edit View Insert Format Tools Data Window Help 26 % Mer. 31 août 10:36:13 identite.csv

Liberation Sans 10 B I U T + - = Code postal

| J | K | L | M | N | O | |
|----|-----------------------|-------|-------------|----------------------------------|-------------------------------------------------|-----------------------------------|
| 1 | Code postal | Ville | Téléphone | Adresse Mail | Nom de la personne à contacter en cas d'urgence | Prénom de la personne à contacter |
| 2 | 93200 Saint-Denis | | 33123456789 | jean.dupont@yahoo.com | DUPONT | Bernadette |
| 3 | 94300 Vincennes | | 33148484848 | tacou.nora@gmail.com | TACOU | Nadia |
| 4 | 91300 Massy | | 33190807060 | zwagr@hotmail.fr | ZWAGROSKY | Serguel |
| 5 | 77700 Coupvray | | 33160102030 | delpoulet@lapost.net | DUPOULET | Marcel |
| 6 | 92400 Courbevoie | | 33156575859 | michel.dusoleil@yahoo.fr | DUSOLEIL | Michel |
| 7 | 78200 Mantes-la-Jolie | | 33180509030 | robin.charles@orange.fr | CHARLES | Oliver |
| 8 | 95400 Arnouville | | 33130354045 | francoise.durant@free.fr | DURANT | Gabriel |
| 9 | 59000 Lille | | 33340506070 | elkebir@alice-adsl.fr | ELKEBIR | Yazid |
| 10 | 75001 Paris | | 33140506080 | vincent.simon@hotmail.com | SIMON | Dominique |
| 11 | 76000 Rouen | | 33235507090 | patricia.lola@orange.com | PORCELET | Raphaël |
| 12 | 93000 Bobigny | | 33148555555 | Patricia.charles-barbier@free.fr | BARBIER | Richard |
| 13 | | | | | | |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | | | | | | |
| 17 | | | | | | |
| 18 | | | | | | |
| 19 | | | | | | |
| 20 | | | | | | |
| 21 | | | | | | |
| 22 | | | | | | |
| 23 | | | | | | |
| 24 | | | | | | |

identite Sheet 1 of 1 Default Sum=0 100% 100%

LibreOffice File Edit View Insert Format Tools Data Window Help 26 % Mer. 31 août 10:36:24 identite.csv

Liberation Sans 10 B I U T + - = Prénom du partenaire extérieur

| S | T | U | V | Commentaires |
|----|--------------------------------|-----------------------------------|--------------------------------------|----------------------------------|
| 1 | Prénom du partenaire extérieur | Téléphone du partenaire extérieur | Adresse Mail du partenaire extérieur | Fonction du partenaire extérieur |
| 2 | Gérard | 33158102030 | gerard.lapeyre@orange.fr | Médecin traitant |
| 3 | Pierre | 33148454749 | pierre.lefebvre@laposte.net | Ophtalmologiste |
| 4 | Marie-France | 33191909090 | mf.duret@gmail.com | ORL |
| 5 | Amadou | 33160708090 | amadou.cisse@wanadoo.fr | Cancérologue |
| 6 | Amandine | 331585858958 | amandine.lepont@hotmail.com | Allergologue |
| 7 | Patrick | 33180907050 | samirpatrick@free.fr | Psychanalyste |
| 8 | Eric | 33130303030 | saideric@aposte.net | Cardiologue |
| 9 | Hamady | 33149404040 | hamady.dosso@wanadoo.com | Kinésithérapeute |
| 10 | Bernard | 33141424344 | bernard.simon@hotmail.com | Orthophoniste |
| 11 | Henri | 33144444444 | henri.michelet@yahoo.com | Ergothérapeute |
| 12 | Aurélien | 33145454545 | aurelien.dupres@wanadoo.com | Auxiliaire de vie |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |

identite Sheet 1 of 1 Default Sum=0 100% 100%

Annexe L : captures d'écran des fichiers departements et statut pour remplir des listes déroulantes de certains écrans de l'application

