



TELECOM NANCY

RAPPORT DU PROJET DE PROJET PLURIDISCIPLINAIRE D'INFORMATIQUE
INTÉGRATIVE



RACHEL BACZYNSKI
ALEXANDRE DHENIN
ARNO RAMM
PROMOTION 2023

Responsable du module :
SÉBASTIEN DA SILVA
GERALD OSTER

30 Mars 2021 — 10 Juin 2021

Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction | 3 |
| 1.1 | Présentation du projet | 3 |
| 1.2 | Organisation du rapport | 3 |
| 1.3 | Précisions légales | 4 |
| 2 | État de l'art | 5 |
| 2.1 | Le concours Mines-Télécom | 5 |
| 2.1.1 | Côté concours | 5 |
| 2.1.2 | Côté candidat | 6 |
| 2.2 | Base de données | 6 |
| 2.2.1 | Troisième forme normale | 6 |
| 2.2.2 | Contraintes d'intégrité | 6 |
| 3 | Gestion de Projet | 8 |
| 3.1 | Organisation de l'équipe projet | 8 |
| 3.2 | Analyse du projet | 8 |
| 3.2.1 | Matrice SWOT | 8 |
| 3.2.2 | Objectif SMART | 9 |
| 3.2.3 | Organisation du projet - Diagrammes de Gantt | 10 |
| 3.3 | Comptes-rendus de réunions | 13 |
| 3.3.1 | Réunion du 30 mars 2021 | 13 |
| 3.3.2 | Réunion du 04 avril 2021 | 14 |
| 3.3.3 | Réunion du 11 avril 2021 | 15 |
| 3.3.4 | Réunion du 27 avril 2021 | 16 |
| 3.3.5 | Réunion du 02 mai 2021 | 17 |
| 3.3.6 | Réunion du 09 mai 2021 | 18 |
| 3.3.7 | Réunion du 16 mai 2021 | 19 |

| | | |
|----------|---|-----------|
| 3.3.8 | Réunion du 23 mai 2021 | 20 |
| 3.3.9 | Réunion du 28 mai 2021 | 21 |
| 3.3.10 | Réunion du 06 juin 2021 | 22 |
| 3.3.11 | Réunion du 10 juin 2021 | 23 |
| 4 | Implémentation de la base de données | 24 |
| 4.1 | Schéma relationnel | 24 |
| 4.2 | Implémentation de la base de données | 25 |
| 4.3 | Remplissage de la base de données | 26 |
| 4.3.1 | Utilisation des fichiers .xlsx et .csv | 26 |
| 4.3.2 | Extrapolation du nombre de places par école | 26 |
| 4.3.3 | Non utilisation de certaines données | 26 |
| 4.4 | Tests | 26 |
| 5 | Application : site internet | 28 |
| 5.1 | Implémentation du site | 28 |
| 5.2 | Ensemble des requêtes | 29 |
| 5.3 | Visuels et graphiques | 29 |
| 5.4 | Fonctionnalités | 31 |
| 6 | Bilan | 32 |
| 6.1 | Bilan global du projet | 32 |
| 6.1.1 | Bilan individuel de Rachel Baczynski | 32 |
| 6.1.2 | Bilan individuel de Alexandre Dhenin | 33 |
| 6.1.3 | Bilan individuel de Arno Ramm | 33 |
| 6.2 | Travail réalisé | 34 |

Chapitre 1

Introduction

1.1 Présentation du projet

Ce projet est réalisé dans le cadre du module Projet Pluridisciplinaire d'Informatique Intégrative de la première année d'école d'ingénieurs TELECOM-Nancy. Ce dernier avait pour but de nous pousser à concevoir une base de données, à manipuler cette dernière pour finalement choisir un ou des langage(s) appris en cours d'année afin de rendre ces données consultables depuis une interface de notre choix.

Après nous être concertés, nous avons choisi de mettre en place le projet sous forme d'une interface web, Python et plus particulièrement sa librairie Flask étaient donc les outils de prédilection pour cela. Le projet ne nécessitait pas l'utilisation de logiciels spécifiques, simplement d'un IDE propre à chacun ainsi que d'un logiciel permettant de consulter les données dans la base de données, comme par exemple SQLiteStudio. Nous avons souhaité respecter l'esprit du concours au maximum et avons donc ajouté plusieurs fonctionnalités telles que la présence d'un système de connexion pour les étudiants à l'aide de leur numéro de candidat, ou encore la possibilité de créer un compte pour un personnel administratif afin qu'il puisse consulter les données du concours.

1.2 Organisation du rapport

Dans ce rapport se trouvent tout d'abord un état de l'art concernant le principe du concours Mines-Télécom, ainsi qu'une courte partie concernant les bases de données. Le chapitre suivant est consacré à la gestion de projet : ses différents supports, les conclusions que nous pouvons en tirer et les comptes-rendus de réunion.

Les quatrième et cinquième chapitres développent les deux parties principales de ce projet : la conception et l'implémentation de la base de données ainsi que son utilisation dans les fonctionnalités de notre site.

Le dernier chapitre présente quant à lui le bilan du projet, à savoir le bilan global et les bilans personnels de chaque membre de l'équipe.

1.3 Précisions légales

Ce projet est destiné à un usage purement scolaire, ainsi les images présentes ne sont pas destinées à la publication et ne sont pas toutes libres de droit.

Cependant, ce caractère strictement scolaire nous autorise à les inclure en accord avec :

- Code civil : articles 7 à 15, article 9 : respect de la vie privée
- Code pénal : articles 226-1 à 226-7 : atteinte à la vie privée
- Code de procédure civile : articles 484 à 492-1 : procédure de référé
- Loi n78-17 du 6 janvier 1978 : Informatique et libertés, Article 38

Chapitre 2

État de l'art

2.1 Le concours Mines-Télécom

2.1.1 Côté concours

Le concours commun Mines-Télécom est l'un des concours pour l'entrée dans les écoles d'ingénieurs françaises. Il est ouvert aux élèves de classe préparatoire aux grandes écoles qui ont suivi un des cursus parmi les suivants : Mathématiques et physique (MP), Physique et chimie (PC), Physique et Sciences de l'Ingénieur (PSI), Physique et Technologie (PT), Technologie et Sciences Industrielles (TSI) et Adaptation Technicien Supérieur (ATS). Le concours est divisé en deux parties : les épreuves écrites puis les épreuves orales.

Il utilise les épreuves du concours commun Mines-Ponts comme banque de notes pour les épreuves écrites des filières MP, PSI, PC et PT, celle du concours Centrale-Supélec pour la filière TSI et celle du concours ATS pour la filière ATS.

A la suite des épreuves écrites, un classement entre les candidats est établi. Une portion d'entre eux est sélectionnée, ces candidats sont alors dits admissible et peuvent participer à la suite du concours : les épreuves orales. Le concours commun Télécom organise deux types d'épreuves orales :

- L'oral Mines-Télécom à l'attention des candidats des filières MP, PSI, PC, PT (permettant d'intégrer les écoles de la banque du concours) ;
- L'oral Mines-Télécom spécifique à l'attention de tous les candidats souhaitant s'y inscrire (il permet d'intégrer des écoles spécifiques).

Pour les filières TSI et ATS, le concours Mines-Télécom utilise les notes des épreuves orales de leurs concours respectifs.

A la suite de ces épreuves orales, les écoles du concours Mines-Télécom sélectionnent un nombre de candidats déterminé selon la filière : les candidats sélectionnés sont alors dits admis.

2.1.2 Côté candidat

Les candidats inscrits aux concours Mines-Télécom passent donc une série d'épreuves écrites puis, si ils sont admissibles, une série d'épreuves orales. Une fois cela terminé, ils établissent une liste de vœux ordonnée (un vœu représentant un souhait pour une école donnée). Les candidats sont alors admis si une des écoles dans leur liste de vœux les a acceptés. Ils sont admis dans l'école de plus haut rang (dans leur liste de vœux) les ayant acceptés. Si aucune des écoles de leur liste ne les ont acceptés, les candidats sont alors éliminés du concours.

2.2 Base de données

2.2.1 Troisième forme normale

Tout au long de la conception du schéma relationnel, nous avons également veillé à ce que notre base de données soit en **troisième forme normale**.

Dans une base de données relationnelle, une forme normale définit un type de relation particulier entre les différentes entités de la base de données. Cela se traduit par un ensemble de propriétés à respecter, et a pour but d'éviter la redondance des données et l'incohérence [5].

Les formes normales s'emboîtent les unes dans les autres, c'est à dire que la n ième forme normale est une $(n-1)$ ième forme normale avec l'ajout d'une ou plusieurs propriétés.

La troisième forme normale respecte quant à elle les propriétés suivantes :

- **Propriété d'atomicité** : les clés et les attributs ne doivent pas avoir une valeur atomique. En pratique, cela signifie qu'on ne peut avoir de NULL ni de tableaux dans notre base de données ;
- Un attribut non clé ne dépend pas d'un ou plusieurs attributs ne participant pas à la clé [5].

Dans la pratique, notre base de données ne respecte pas complètement ces différentes propriétés : en effet, nous avons besoin de garder certaines valeurs potentiellement nulles afin de permettre un remplissage de la base de données efficace.

Par ailleurs, la table ATS que nous avons implémentée va à l'encontre de ses propriétés puisqu'elle est en quelques sortes une "vue" de la table Eleve. Nous avons fait ce choix afin de garder un maximum de cohérence par rapport à la réalité du concours et cloisonner un maximum nos données lorsqu'il était possible. En effet, la filière ATS est une filière "à part", ce qui explique pourquoi les ATS possèdent leur propre table.

2.2.2 Contraintes d'intégrité

Les contraintes d'intégrité représentent un ensemble de règles et propriétés que doivent respecter les champs d'une base de données afin de garantir la cohérence, la pertinence

et la validité de ces données (et de l'implémentation de celle-ci). Il existe trois contraintes d'intégrité essentielles [4] :

- **La contrainte de domaine** : chaque attribut doit prendre une valeur dans le domaine de valeurs déterminé ;
- **La contrainte de relation** : chaque relation du modèle relationnel doit être identifiée par une clé primaire unique et non nulle ;
- **La contrainte de référence (ou contrainte d'intégrité référentielle)** : les clés étrangères (qui permettent de relier les tables et relations entre elles) sont incluses dans le domaine de valeur de la clé primaire dont elles sont issues. On définit donc cette clé de façon à ce qu'elle soit de même taille et de même type que la clé primaire qui la détermine.

Il sera expliqué dans la partie 4.1 (Implémentation de la base de données : Schéma relationnel) comment la base de données que nous avons implémentée respecte ces différentes contraintes.

Chapitre 3

Gestion de Projet

3.1 Organisation de l'équipe projet

L'équipe projet a décidé d'adopter la méthode agile SCRUM pour la réalisation de ce projet. Alexandre Dhenin a été désigné Scrum Master par l'ensemble de l'équipe tandis que les autres membres constituent le reste de l'équipe de développement.

L'équipe se réunissait majoritairement les dimanche à 18h sur Discord. Lors de ces réunions, il était question du suivi du projet, de la définition des charges de travail et des objectifs pour le sprint de la semaine suivante. En raison de la taille du projet les mêlées n'ont pas nécessairement été quotidiennes, elles se faisaient la plupart du temps en distanciel à l'aide de l'outil messenger.

Nous avons utilisé les outils fournis tels que Gitlab et Overleaf permettant la collaboration en simultané de tous les membres de l'équipe projet.

Les membres ont cependant utilisés des IDE différents pour la partie algorithmique.

3.2 Analyse du projet

3.2.1 Matrice SWOT

Nous avons fait une matrice SWOT lors de la première réunion, afin de bien identifier les caractéristiques internes (strengths, weaknesses) et externes (opportunities, threats) du projet et de l'équipe projet. Celle-ci a été quelque peu modifiée durant la semaine qui suit à mesure que la réflexion avançait. La voici :

| | |
|--|---|
| <p>Strengths</p> <ul style="list-style-type: none"> - Bonne communication - Arno : expérience en WebBD, Java, C - Alexandre, Rachel : bonne expérience du premier projet - Arno : rigueur - Assiduité | <p>Weaknesses</p> <ul style="list-style-type: none"> - Alexandre, Rachel : peu d'expérience en informatique - Membres de l'équipe-projet ne se connaissent pas du tout |
| <p>Opportunities</p> <ul style="list-style-type: none"> - Aide possible venant du corps enseignant de l'école ou des 2A/3A - Originalité : se démarquer des autres en produisant une réalisation originale - Consolidation des connaissances déjà acquises en informatique | <p>Threats</p> <ul style="list-style-type: none"> - Incertitudes face à la situation sanitaire - Contraintes de temps - Projet commun à toute la promotion : attention à ne pas s'inspirer - Sujet assez vague - Tous les membres de l'équipe-projet dans des groupes différents : difficulté à trouver des créneaux pour travailler ensemble |

TABLE 3.1 – Matrice SWOT

3.2.2 Objectif SMART

Nous avons fait notre objectif SMART à notre première réunion, en même temps que la matrice SWOT. Le voici :

| Critère | Application au projet | Indicateur |
|------------------------|---|--|
| Spécifique | L'objectif est défini clairement. | Sujet ambiguë mais réunions de l'équipe projet le préciseront. |
| Mesurable | L'objectif est mesurable par des indicateurs chiffrés ou des livrables. | Rapport LaTeX avec toutes les sections précisées dans le sujet, soutenance. Deadline bien définie. |
| Atteignable | L'objectif doit être motivant sans être décourageant et doit apporter un plus par rapport au lancement du projet. | Chacun est d'accord avec les décisions prises ainsi qu'avec la répartition des tâches. |
| Réaliste | L'objectif est réaliste au regard des compétences et de l'investissement de l'équipe projet. | Objectifs réalisables de semaine en semaine sur des domaines connus ou abordables. |
| Temporellement définie | Il doit être inscrit dans le temps, avec une date de fin et des jalons. | La date de départ est le 30 mars, celle de rendu est le 10 juin. |

TABLE 3.2 – Objectif SMART

3.2.3 Organisation du projet - Diagrammes de Gantt

Le projet se déroule sur une période donnée, il est indispensable de planifier son déroulement dans le temps. Le projet a été découpé en plusieurs étapes ce qui facilite l'organisation.

A cet effet, deux diagrammes de Gantt ont été réalisés : un prévisionnel et un réel. Le diagramme de Gantt prévisionnel a été réalisé en amont dans les premières semaines du projet. Le diagramme de Gantt réel a été finalisé la dernière semaine du projet.

La comparaison de ces deux diagrammes indique que nous avons globalement bien réussi à suivre notre répartition de tâches initiales. Les seules différences notables sont les tests qui ont démarré plus tard qu'initialement prévu et l'implémentation du site qui a, au contraire, démarré plus tôt que prévu. Voici les deux diagrammes de Gantt :

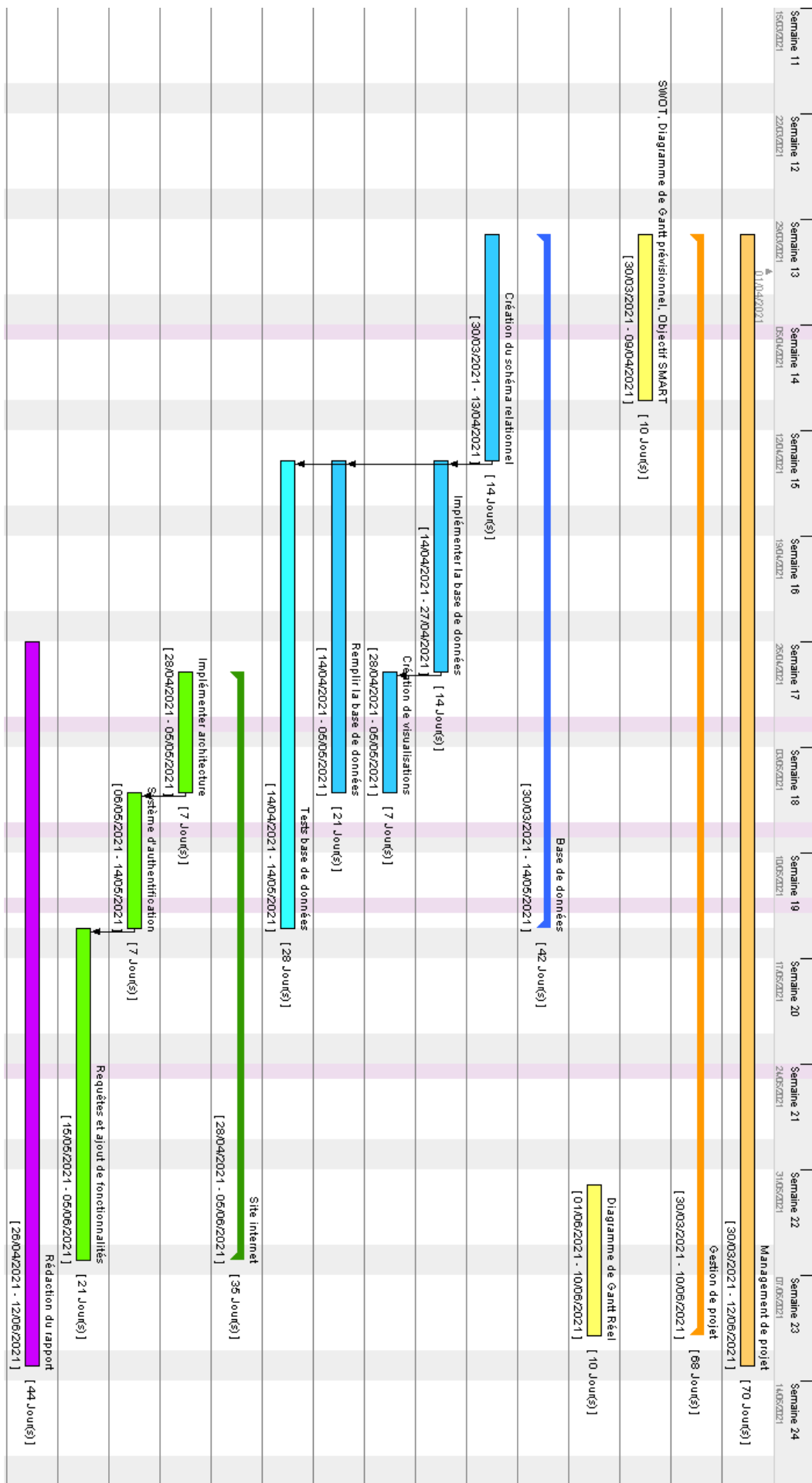


FIGURE 3.1 – Diagramme de Gantt prévisionnel (logiciel : Gantt Project)

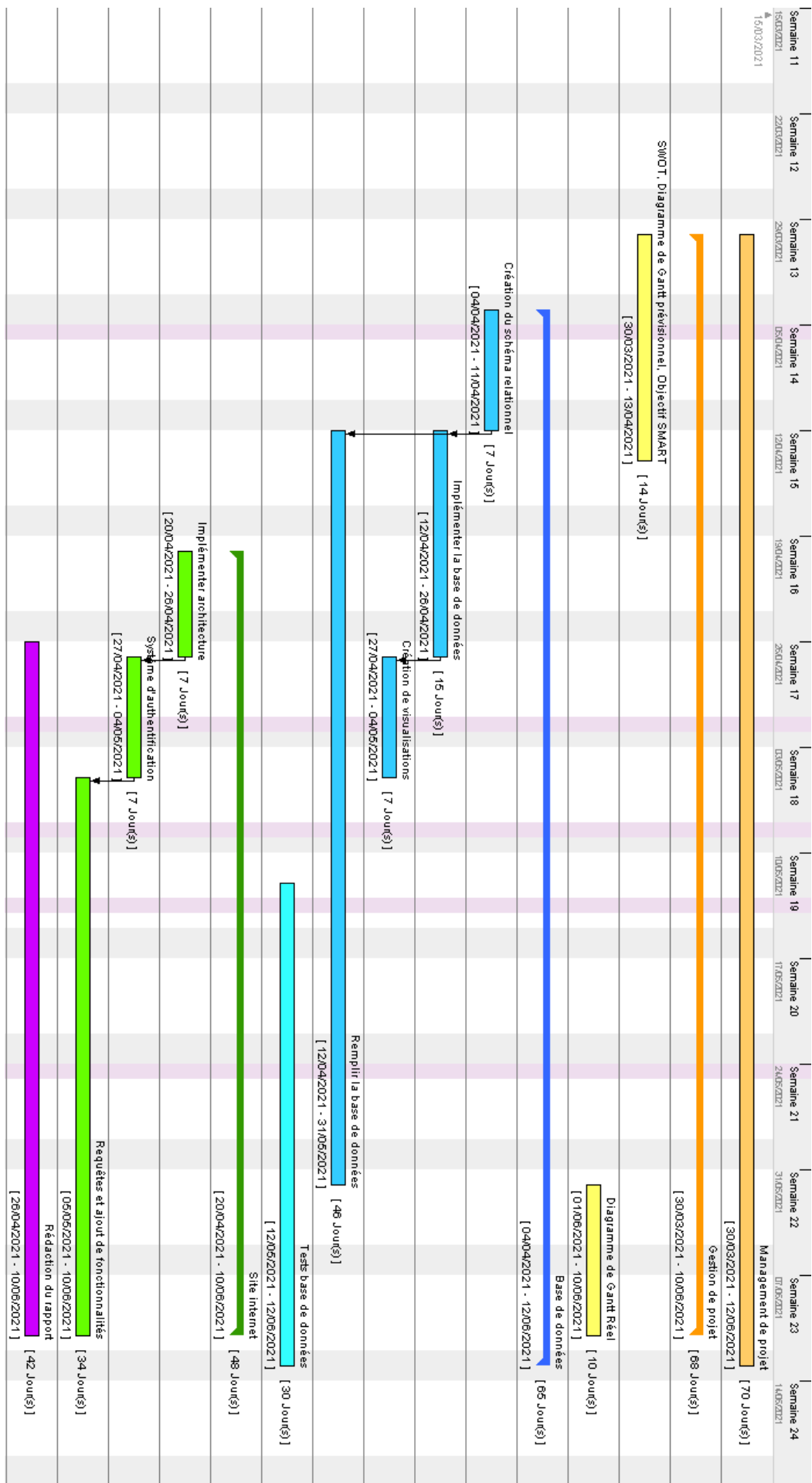


FIGURE 3.2 – Diagramme de Gantt réel (logiciel : Gantt Project)

3.3 Comptes-rendus de réunions

Pour éviter l'effet tunnel, nous avons tenu à réaliser des comptes rendus de chaque réunion afin que chacun puisse suivre ses objectifs à court terme.

3.3.1 Réunion du 30 mars 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|--|
| Équipe projet : - Arno RAMM - Alexandre DHENIN - Rachel BACZYNSKI | Date : 30 mars 2021 Durée : 1h Horaire : 14h |

Ordre du jour :

- Mise en place des mécanismes de gestion de projet
- Répartition des tâches

Bilan du sprint de la semaine :

Première semaine !

Décisions prises / Todo-List :

1. Familiarisation avec le sujet (lecture, recherche sur les différentes possibilités)
2. Poser questions par mail en cas d'incompréhensions

Date de la prochaine réunion : 06 avril 2021

3.3.2 Réunion du 04 avril 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|--|
| Équipe projet : - Arno RAMM - Alexandre DHENIN - Rachel BACZYNSKI | Date : 04 avril 2021 Durée : 30min Horaire : 17h30 |

Ordre du jour :

- Répartition des tâches

Bilan du sprint de la semaine :

Nous avons obtenu les réponses à nos questions ! Chacun a pu étudier le sujet de son côté et en discuter avec les autres membres du groupe.

Décisions prises / Todo-List :

- Création du schéma normalisé de la Base de Données (Arno)
- Développement d'un script d'extraction des données (Alexandre)
- Développement de fonctions de recherche de redondance (Rachel)

Date de la prochaine réunion : 07 avril 2021

3.3.3 Réunion du 11 avril 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|---|--|
| Équipe projet : <ul style="list-style-type: none">- Arno RAMM- Alexandre DHENIN- Rachel BACZYNSKI | Date : 11 avril 2021 Durée : 30min Horaire : 17h30 |

Ordre du jour :

- Derniers débats concernant le schéma relationnel
- Changement dans les technologies utilisées (extraction des données en Python et non plus en C)

Bilan du sprint de la semaine :

Le schéma relationnel est quasiment terminé, il reste quelques incohérences à supprimer.

Décisions prises / Todo-List :

- Fin du schéma relationnel (Arno)
- Début des scripts en Python afin de traiter les données (Alexandre + Rachel)
- Vérifier la cohésion du schéma relationnel avec l'esprit du concours (Alexandre)

Date de la prochaine réunion : 27 avril 2021

3.3.4 Réunion du 27 avril 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|---|--|
| Équipe projet : <ul style="list-style-type: none">- Arno RAMM- Alexandre DHENIN- Rachel BACZYNSKI | Date : 27 avril 2021 Durée : 45min Horaire : 14h30 |

Ordre du jour :

- Répartition des tâches
- Changements mineurs concernant la base de données
- Point sur l'architecture du site

Bilan du sprint de la semaine :

Chaque participant a pu bien avancer de son côté, la réunion a servi à rectifier quelques incohérences de la base de données qui étaient passées inaperçues, elle a également permis de s'entraider en terme d'utilisation du SQL.

Décisions prises / Todo-List :

- Développement d'un système de création de compte et de connexion (Arno)
- Avancement dans le script de remplissage de la base de données (Alexandre)
- Fin de la rédaction des requêtes (Rachel)

Date de la prochaine réunion : 02 mai 2021

3.3.5 Réunion du 02 mai 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|---|--|
| Équipe projet : <ul style="list-style-type: none">- Arno RAMM- Alexandre DHENIN- Rachel BACZYNSKI | Date : 02 mai 2021 Durée : 30min Horaire : 18h00 |

Ordre du jour :

- Discussions sur l'ajout des ATS au sein de la base de données
- Discussions autour de nouvelles requêtes à implémenter sur le site
- Mise à jour de la gestion de projet

Bilan du sprint de la semaine :

La plupart des requêtes qui seront présentes sur le site ont été écrites et il ne reste plus qu'à les intégrer. Le site est en ligne, le système de création de compte et d'authentification fonctionne et le style a été ajouté. Il ne reste plus qu'à attendre la fin du script de remplissage des données afin de pouvoir vraiment tester toutes les fonctionnalités.

Décisions prises / Todo-List :

- Implémentation des nouvelles requêtes sur le site (Arno)
- Modifications dans le schéma relationnel + Python (Arno)
- Mise à jour de la gestion de projet (Rachel)
- Tri dans les requêtes (Rachel)
- Fin du script de remplissage des données (Alexandre)

Date de la prochaine réunion : 09 mai 2021

3.3.6 Réunion du 09 mai 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|--|
| Équipe projet : - Arno RAMM - Alexandre DHENIN - Rachel BACZYNSKI | Date : 09 mai 2021 Durée : 30min Horaire : 18h00 |

Ordre du jour :

- Point sur le projet et répartition des tâches
- Discussions autour de changements à apporter au site web
- Discussions concernant le développement d'un script générant dynamiquement des statistiques

Bilan du sprint de la semaine :

Quasiment toutes les fonctionnalités souhaitées ont été implémentées sur le site. Nous avons fait le point et relu le sujet afin de savoir si certaines consignes avaient été oubliées ; cela nous a permis d'avoir l'idée de pouvoir exporter les pages au format PDF.

Décisions prises / Todo-List :

- Changements dans certaines vues (Arno)
- Renseignement sur l'exportation en PDF (Arno)
- Script permettant de créer des graphiques statistiques (Alexandre)

Date de la prochaine réunion : 16 mai 2021

3.3.7 Réunion du 16 mai 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|--|
| Équipe projet : - Arno RAMM - Alexandre DHENIN | Date : 16 mai 2021 Durée : 60min Horaire : 18h00 |

Ordre du jour :

- Mise au point sur les fonctionnalités manquantes du site
- Nécessité d'une mise au point concernant la répartition des tâches et l'équilibre de chacun

Bilan du sprint de la semaine :

La semaine était chargée en cours et en partiels, et deux membres du groupe ont profité du weekend de pentecôte pour faire d'autres activités ; le projet n'a donc pas vraiment progressé cette semaine. Arno a pu continuer d'avancer sur les dernières fonctionnalités du site mais a été bloqué sur un point qui nécessitait un éclaircissement d'Alexandre. Rachel n'a pas pu assister à cette réunion.

Décisions prises / Todo-List :

- Nécessité d'une réunion concernant la répartition et la charge du travail (initiative Arno + Alexandre)
- Implémentation et fin des fonctionnalités sur le site d'ici la prochaine réunion (Arno)
- Création de scripts de test de complétude (Rachel)
- Vérifier les attendus du sujet, identifier les contraintes d'intégrité (Rachel)
- Optimiser le script de remplissage de la BDD (Alexandre)
- Changement mineurs dans le schéma de la BDD en Python (Alexandre)

Date de la prochaine réunion : 23 mai 2021

3.3.8 Réunion du 23 mai 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|--|
| Équipe projet : - Arno RAMM - Alexandre DHENIN - Rachel BACZYNSKI | Date : 23 mai 2021 Durée : 30min Horaire : 19h00 |

Ordre du jour :

- Mise au point sur l'organisation du groupe

Bilan du sprint de la semaine :

Le site a beaucoup avancé cette dernière semaine mais nous nous sommes rendus compte que la répartition des tâches n'était pas assez équitable et qu'il fallait effectuer un ré-équilibre le plus rapidement possible, c'est donc le sujet du jour.

Décisions prises / Todo-List :

- Création de scripts de tests de complétude / cohérence de la BDD (Rachel)
- Rédaction de la partie "démonstration de la 3FN" (Rachel)
- Amélioration de la rapidité du script de création de la BDD (Alexandre)
- Correction de bugs, amélioration des pages et rajout d'informations (Arno)

3.3.9 Réunion du 28 mai 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|---|--|
| Équipe projet : <ul style="list-style-type: none">- Arno RAMM- Alexandre DHENIN- Rachel BACZYNSKI | Date : 28 mai 2021 Durée : 80min Horaire : 19h00 |

Ordre du jour :

- Discussions sur le thème du site
- Dernière fonctionnalités à implémenter

Bilan du sprint de la semaine :

Le site est quasiment terminé, il ne reste plus qu'à réunir le groupe pour se mettre d'accord sur le thème visuel du site. (Arno étant daltonien, le site était resté en noir et blanc jusqu'ici afin de faciliter le contraste).

Décisions prises / Todo-List :

- Changer le style du site et rajouter des images (Arno)
- Continuer le rapport et les tests (Rachel)
- Corrections mineures au niveau de la gestion des ATS et rajout de quelques informations manquantes dans la BDD (Alexandre)
- Création d'images à insérer sur le site (Alexandre)

Date de la prochaine réunion : 6 juin 2021

3.3.10 Réunion du 06 juin 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|--|---|
| Équipe projet : - Arno RAMM - Alexandre DHENIN - Rachel BACZYNSKI | Date : 06 juin 2021 Durée : 30min Horaire : 19h00 |

Ordre du jour :

- Dernières fonctionnalités, correction de bugs

Bilan du sprint de la semaine :

Le site présente quelques bugs qu'il faut corriger avant le rendu. Nous avons également reçu, depuis la dernière réunion, des consignes un peu plus précises concernant les attendus; nous avons donc décidé de rajouter une fonctionnalité au site.

Décisions prises / Todo-List :

- Rédaction de certaines parties du rapport (Rachel)
- Correction de bugs (Arno)

Date de la prochaine réunion : 10 juin 2021

3.3.11 Réunion du 10 juin 2021

| Motif : Suivi du projet | Lieu : Plateforme vidéo Discord |
|---|---|
| Équipe projet : <ul style="list-style-type: none">- Arno RAMM- Alexandre DHENIN- Rachel BACZYNSKI | Date : 10 juin 2021 Durée : 30min Horaire : 19h00 |

Ordre du jour :

— Fin de la rédaction du rapport !

Bilan du sprint de la semaine :

Fin de la rédaction du rapport !

Décisions prises / Todo-List :

Date de la prochaine réunion : Le projet est terminé!

Chapitre 4

Implémentation de la base de données

Comme précisé en introduction, notre projet nécessitait une base de données dans laquelle seraient importées les données de manière automatisée, à partir de fichiers excels. Le sujet ne précisait pas de contraintes, si ce n'est de respecter une certaine forme normale, ainsi que d'éviter au maximum la redondance et d'assurer la cohérence des données. La première étape de cette conception était donc la réalisation d'un schéma relationnel.

4.1 Schéma relationnel

Afin de réaliser le schéma relationnel, nous n'avons pas utilisé de méthodes spécifiques mais simplement les entêtes des fichiers excels et l'expérience d'Arno venant de DUT. Le schéma a été réalisé progressivement en parcourant les fichiers un par un, et était mis à jour et partagé avec les autres membres du groupe sur le site dbdiagram.io.

Il a été réalisé en moins d'une semaine mais a beaucoup évolué durant l'implémentation de ce dernier en Python, l'équipe se rendant compte au fur et à mesure que certaines tables pouvaient être simplifiées, ou bien encore qu'il manquait certaines colonnes. Au total, plus d'un mois aura été nécessaire afin de trouver le schéma qui convenait presque parfaitement à nos besoins sur le site. Nous avons fait le choix d'utiliser un grand nombre de tables (plus de vingt) simplement afin de séparer les données qui pouvaient l'être pour maintenir une cohérence mais également pour optimiser la taille de la base de données et la rapidité des requêtes.

Par ailleurs, nous avons bien tenu compte des **contraintes d'intégrité** pour réaliser la structure de notre base de données, que ce soit pour :

- **La contrainte de domaine** : chacun des attributs de notre base de données est limité aux valeurs qu'il peut recevoir. Par exemple, les notes des candidats sont toutes comprises entre 0 et 20, et les numéros de téléphone des candidats respectent tous le même format.
- **La contrainte de relation** : chacune des relations de notre base de données est identifiées par une clé primaire qui est **unique** et **non nulle** de manière intrinsèque.

- **La contrainte de référence** : en raison du grand nombre de tables dans notre base de données, nous avons utilisé beaucoup de clés étrangères. Celles-ci nous permettent d'assurer une cohérence entre les données puisque ces dernières ne peuvent pas présenter de différences entre deux tables lorsqu'elles se référencent. Nous n'avons pas jugé utile d'utiliser des "gâchettes" ou "déclencheurs" : en effet, nous n'avons pas besoin de mettre à jour les informations dans notre base de données depuis le site (ce qui aurait pu causer des problèmes de cohérence).

Nous avons aussi dû revoir certaines contraintes pendant le développement, comme par exemple mettre les notes en **float** plutôt qu'en **int** pour pouvoir prendre en compte les notes non entières.

4.2 Implémentation de la base de données

Le schéma de la base de données a été implémenté en Python à l'aide de la librairie sqlite3. Cette dernière nous a permis d'ouvrir un fichier de base de données (ou de le créer s'il n'existe pas), d'y créer des tables (si elles n'existent pas) respectant les colonnes et les contraintes du schéma relationnel, puis d'utiliser la connexion à la base de données afin d'exécuter des requêtes SQL directement dans le code. Toutes les méthodes et requêtes permettant de créer la base de données et ses tables ont été regroupées dans un fichier nommé "DBCreation.py" présent dans le répertoire principal du projet.

```
connexion.execute('CREATE TABLE IF NOT EXISTS NATIONALITE
([id] INTEGER PRIMARY KEY AUTOINCREMENT,
[libelle] text NOT NULL)')

connexion.execute('CREATE TABLE IF NOT EXISTS QUALITE
([id] INTEGER PRIMARY KEY AUTOINCREMENT,
[libelle] text NOT NULL)')

connexion.execute('CREATE TABLE IF NOT EXISTS PAYS
([code_pays] INTEGER PRIMARY KEY NOT NULL,
[nom] text NOT NULL)')

connexion.execute('CREATE TABLE IF NOT EXISTS ETABLISSEMENT
([rne] TEXT PRIMARY KEY NOT NULL,
[nom_etab] text NOT NULL,
[type_etab] text NOT NULL,
[code_postal] integer NOT NULL,
FOREIGN KEY (code_postal)
REFERENCES VILLE (code_postal))')

connexion.execute('CREATE TABLE IF NOT EXISTS VILLE
([code_postal] INTEGER PRIMARY KEY NOT NULL,
[nom_ville] text NOT NULL,
[pays_ville] integer NOT NULL,
FOREIGN KEY (pays_ville)
REFERENCES PAYS (code_pays))')
```

FIGURE 4.1 – Extrait du code de DBcreation.py

4.3 Remplissage de la base de données

4.3.1 Utilisation des fichiers .xlsx et .csv

La prochaine étape a été de développer un script qui permet d'importer les données des différents fichiers excels dans la base de données. Pour cela, nous avons tout d'abord procédé à un formatage des fichiers, puisque certains ne sont pas au même format (première colonne d'entête, données décalées de une case vers la droite, même donnée retranscrite de manière différente, etc.). Après cette étape, nous avons pu extraire les données, en les important dans Python à l'aide du module *sqlite3* [2] puis en procédant à des vérifications de cohérence [1](moyennes inférieures à vingt, clé primaire non présente dans la base de données, informations cohérentes entre les tables, etc.). Ce script correspond au fichier *remplissage.bdd.py* qui est stocké dans le répertoire *./utils* du projet.

4.3.2 Extrapolation du nombre de places par école

Une fois ce travail terminé nous nous sommes aperçus qu'il manquait une donnée essentielle dans les fichiers : le nombre de places d'admission par filière et par école. Sans cela nous ne pouvions donner les résultats d'admission pour chaque candidat et plusieurs données implémentées dans notre base de données devenaient "inutiles". Nous avons alors fait le choix d'extrapoler et de donner à chaque école un nombre de places par filière et un nombre de places en file d'attente pour coller au mieux à ce qui existe déjà : le site SCEI. Ainsi, le nombre de places par filière que nous avons donné est proportionnel au nombre de candidats inscrits dans chaque filière. Il est possible que certains candidats ne soient admis dans aucune école, cela est intentionnel et a pour but de coller le mieux possible à la réalité. Une fois tout cela réalisé, nous avons développé un algorithme associant à chaque candidat l'école où il est admissible et les éventuelles écoles où il est en liste d'attente. Ce script correspond au fichier *liste.voeux.scei.py* qui est stocké dans le répertoire *./utils* du projet.

4.3.3 Non utilisation de certaines données

Par ailleurs, certaines données restaient encore incomplètes et ne nous étaient donc pas utiles. Nous avons donc fait le choix de n'implémenter dans la base de données que les données pertinentes. C'est la raison qui nous a conduits à n'utiliser qu'une minorité de candidats ATS car pour la plupart nous n'avions comme seule information leur numéro de candidat.

4.4 Tests

Nous avons réalisé un ensemble de tests dédiés à la base de données. Ceux-ci sont trouvable dans le fichier *Tests/* de notre dépôt gitlab. Ils ont été codés en python, avec l'im-

port de sqlite3 pour récupérer des informations de notre base de données et pandas et os pour la manipulation des excels. Le module pytest a été employé pour effectuer tous les tests, exécuté depuis une fenêtre de commande.

Nous sommes d'abord partis sur des tests qui permettent de garantir que nous avons bien recueilli toutes les données nécessaires des excels et les avons implémentées correctement, sans en perdre ni en altérer aucune. Nous avons commencé par nous assurer que tous les candidats étaient bien présents dans la base de données, avant de vérifier divers éléments comme la présence d'autant de candidats admissibles dans les excels que de candidats admissibles dans notre base de données.

Le principe du code est le suivant : une première boucle portant sur l'ensemble des excels fournis stocke les informations nécessaires dans diverses listes, qui sont elles ensuite parcourues dans des boucles pytest itérant sur les données de la base de données, comme par exemple :

```
def test_codes_candidats():
    for code in candidats_codes:
        a = cur.execute('SELECT can_cod FROM ELEVE WHERE
            can_cod = (?)', (float(code),)).fetchone()
        assert a == (float(code),)
```

Nous nous sommes ensuite rendus compte que tester une base de données était très différent de tester des algorithmes d'analyse (que nous avons pu pratiquer lors de projet(s) précédent(s)). Nous avons donc fait des recherches sur comment tester une base de données.

Nous avons ainsi réalisé que la manière la plus efficace de tester la cohérence de notre base de données reposait sur la manipulation d'excels de bien plus petite taille. L'idée était en effet de créer des excels à la même structure que les excels fournis, remplis de données similaires. Ceux-ci seraient insérés dans une copie de notre base de données avec le même algorithme que celui de notre "réelle" base de données. Ensuite, les données seraient récupérées et stockées dans d'autres excels ensuite comparés aux plus petits excels. Si aucune erreur n'est renvoyée, notre base de données est, à priori, parfaitement cohérente.

Malheureusement, nous n'avons pas pu réaliser ces tests-là par manque de temps et compétence, et avons privilégié les tests précédents.

Pour ce qui est des contraintes d'intégrité de notre base de données, les contraintes de relation et de référence sont vérifiées par construction de notre schéma relationnel, nous avons donc simplement testé la contrainte de domaine.

Ainsi, nous nous sommes par exemple assurés que toutes les notes des candidats étaient bien comprises entre 0 et 20, ou que les noms de villes insérés étaient bien tous des chaînes de caractères.

Chapitre 5

Application : site internet

La base de données permet de stocker toutes les données du concours, la mise en place d'une interface était donc nécessaire afin de pouvoir consulter ces dernières. Comme dit précédemment, nous avons utilisé Python et sa librairie Flask afin de créer un site internet.

Note : toutes les images de ce chapitre proviennent du site internet que nous avons implémenté.

5.1 Implémentation du site

Le site web a été développé en parallèle du script de remplissage de la base de données et de l'écriture des requêtes. Alexandre ayant déjà passé les concours Mines-Telecom, cela nous a guidé afin de réaliser un site se rapprochant le plus possible de la réalité du concours. Nous avons donc décidé de mettre en place un système de connexion au site et de diviser ce dernier en deux parties : une pour les candidats et une pour les personnels administratifs souhaitant consulter les données du concours.

Lors du concours, les candidats doivent se connecter au site de consultation de notes à l'aide de leur numéro de candidat et d'un mot de passe généré aléatoirement qui leur est envoyé par mail. Nous avons implémenté ce système, à la seule différence près que le mot de passe n'est pas envoyé par mail pour des raisons évidentes, mais chaque étudiant se voit directement attribué un mot de passe dans la base de données. En se connectant à l'aide de son numéro de candidat et son mot de passe, l'étudiant a accès à diverses informations qui seront détaillées dans la section suivante. Pour le personnel administratif, il est nécessaire de créer un compte à l'aide d'un e-mail et d'un mot de passe de leur choix. Par le biais de ce couple, ils peuvent se connecter à la plateforme et accéder à des informations concernant les candidats ou des statistiques.

Cette notion d'authentification implique la présence d'une session qui se crée lorsque l'utilisateur se connecte et se détruit lors de la fermeture du site. Pour cela, nous avons utilisé Flask et sa librairie Session qui permettent de réaliser exactement ce que nous recherchions : un tableau associatif est créé lors de la connexion et garde en mémoire le nu-

méro de candidat / l'adresse mail (dans le cas d'un personnel) de la personne connectée. Cette information permet d'effectuer un grand nombre de requêtes et d'afficher toutes les informations souhaitées.

Le site utilise la notion de "templates" proposée par Flask afin de le rendre plus rapide et qu'il n'y ait pas de temps de chargement. Le header du site est chargé dans le template de base nommé "base.html", ce qui a pour effet de ne la charger qu'une fois. Lorsqu'un utilisateur se connecte, la barre de navigation est affichée dynamiquement et des onglets sont créés en fonction de si l'utilisateur est un élève ou un personnel administratif. Il existe plusieurs templates (un par onglet), qui se trouvent dans le dossier "templates".

De nombreuses fonctionnalités sont disponibles sur le site et sont complètement liées aux requêtes réalisées, que nous allons présenter lors de la section suivante.

5.2 Ensemble des requêtes

Toutes les requêtes réalisées ont été écrites dans des fichiers Python portant le nom de la table sur laquelle la requête s'applique (exemple : `eleve.py`). Toutes les requêtes concernant la même table sont regroupées dans ces fichiers et permettent de facilement les importer en tant que module lorsque nous avons besoin d'effectuer une requête.

5.3 Visuels et graphiques

Avoir des données est intéressant mais pour les exploiter il est nécessaire de créer et d'utiliser différents visuels. Certains candidats n'avaient pas de notes ou bien pas de vœux, nous avons alors créé certaines images permettant d'informer l'utilisateur du site de manière didactique.

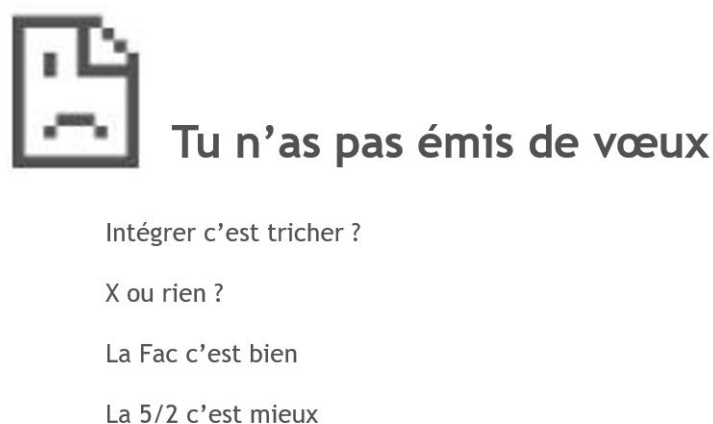


FIGURE 5.1 – Message d'erreur pour un candidat n'ayant pas émis de vœux

Pour un candidat visualisant ses notes, il est intéressant de pouvoir se comparer aux autres candidats. Un graphique en araignée nous a alors semblé particulièrement adapté. Nous en avons développé un dans le script *graph.note.eleve.py* situé dans le dossier utils. Il permet au candidat de visualiser ses notes et de voir la moyenne des candidats de sa filière dans ces mêmes épreuves.

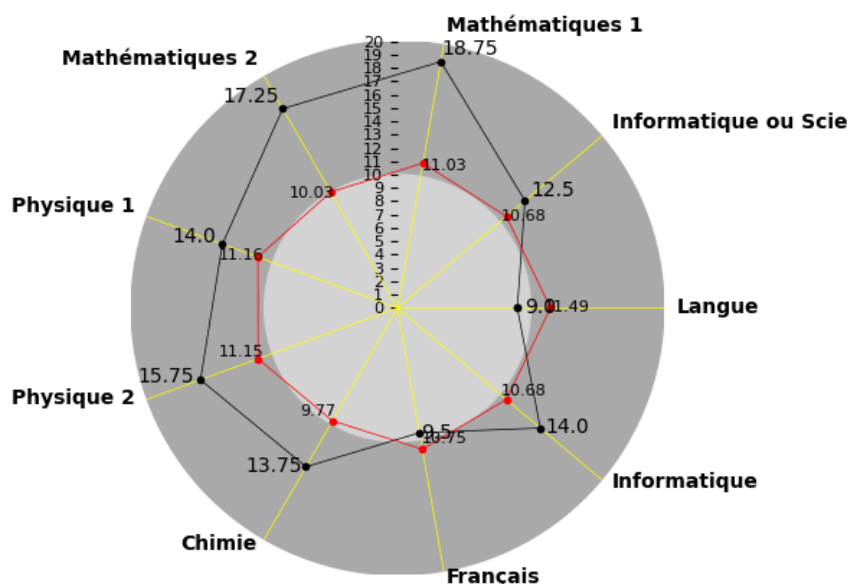


FIGURE 5.2 – Graphique en araignée des notes du candidat 20495 en filière MP

De la même façon pour un administrateur du concours Mines-Télécom, il est intéressant d’avoir une vision globale des notes par matière et par filière.

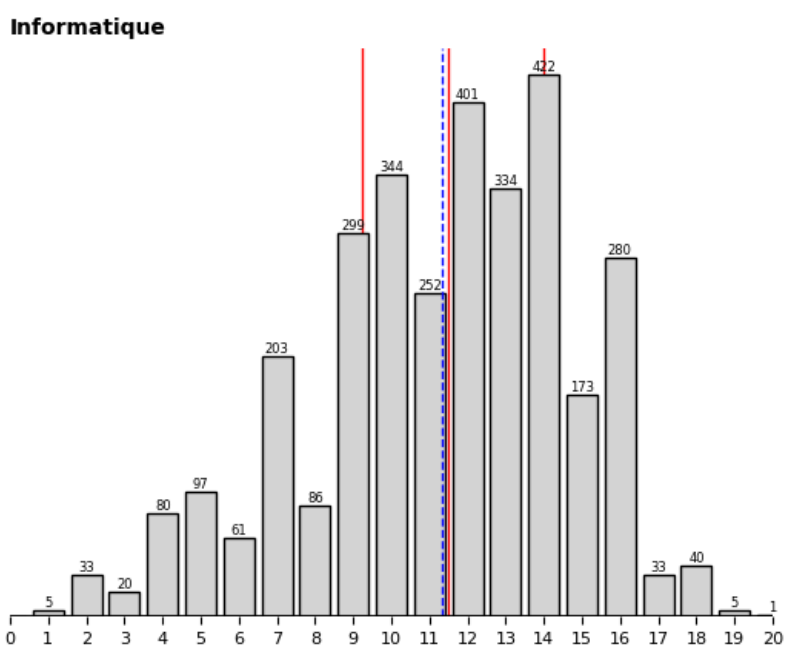


FIGURE 5.3 – Graphique statistique pour l’épreuve d’informatique en filière MP

Nous avons décidé d'implémenter les graphiques présents dans les rapports du concours centrale [3] car ce sont ceux que nous trouvons les plus précis et les plus clairs. Ces graphiques permettent de visualiser les quartiles, la moyenne et globalement quels sont les panels de notes les plus fréquents. Ce script est nommé *graph.note.stats* et est situé dans le dossier ./utils.

Enfin, nous avons ajouté une multitude d'autres visuels permettant de rendre la navigation sur le site plus plaisante.

5.4 Fonctionnalités

Avec cet ensemble de données, nous avons mis au point plusieurs fonctionnalités. Comme expliqué précédemment, notre site web est "divisé" en deux parties. Un côté destiné aux candidats et un autre aux administrateurs du concours Mines-Télécom.

Les administrateurs ont accès aux informations (notes incluses) de tous les candidats, ils peuvent les rechercher par leur numéro de candidat ou en filtrant les candidats par leur filière. Ils ont également accès à l'ensemble des statistiques (moyenne, quartiles, répartition des notes des candidats...) pour chaque épreuve dans une filière donnée. Ils peuvent aussi voir toutes les écoles membre du concours et avoir accès à l'ensemble des admis dans cette école ainsi que le rang du premier et du dernier admis par filière.

Les candidats ont accès à leurs notes, leurs résultats d'admission ainsi qu'à la même partie statistique que les administrateurs du concours.

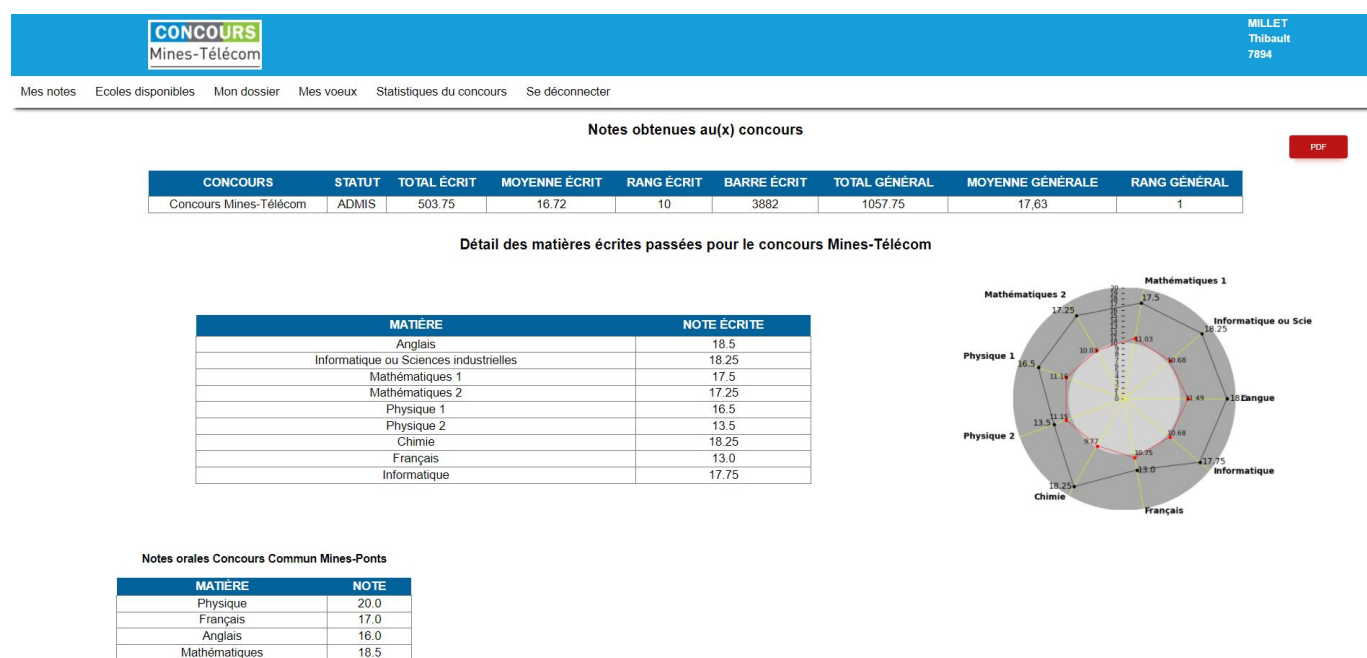


FIGURE 5.4 – Note du candidat numéro 7894

Chapitre 6

Bilan

6.1 Bilan global du projet

Ce projet nous aura permis de développer de nombreuses compétences, notamment en terme de conception de base de données, de création et d'utilisation de requêtes et de développement "pur". Le fait que les groupes aient été formés au hasard a permis de réunir des personnes de différentes études et avec différentes compétences, ce qui nous a permis d'échanger et d'apprendre les uns des autres ! Cette expérience a été enrichissante et nous sera forcément utile pour la suite de nos études puisqu'il s'agit de notre premier projet réalisé de A à Z.

6.1.1 Bilan individuel de Rachel Baczynski

| |
|---|
| Points positifs |
| Je me suis grandement familiarisée avec les différents logiciels informatiques mobilisés, notamment la manipulation de plusieurs outils différents ensemble. J'ai appris à travailler avec des personnes relativement inconnues. |
| Difficultés rencontrées |
| J'ai rencontré des difficultés lors de la manipulation d'excels sur python et de pytest depuis un terminal : il m'a fallu un certain temps de familiarisation avant de pouvoir commencer à réellement programmer une série de tests effectifs. J'ai également connu une courte période durant laquelle j'ai eu une forte baisse de motivation pour le projet. |
| Expérience personnelle |
| Une fois bien impliquée dans le sujet, j'ai beaucoup apprécié travailler sur ce projet ! J'ai aimé me former d'avantage en informatique et me poser des questions quand il y avait des problèmes. Le projet était bien plus diversifié que celui que nous avons déjà fait au premier semestre, et laissait plus de place aux choix : cela nous a poussé à réfléchir et prendre des décisions. |

6.1.2 Bilan individuel de Alexandre Dhenin

| |
|--|
| Points positifs |
| Une très belle application des bases de données qui m'a permis de me remettre à jours dans la manipulation de ces dernières. Une première initiation à la création d'un site WEB de A à Z. Un travail de groupe toujours aussi stimulant. |
| Difficultés rencontrées |
| Une communication plus difficile car nous ne nous connaissions pas avant le début du projet. Une erreur au début de la conception de l'algorithme de remplissage de la base de donnée qui n'a pas été sans conséquence (j'ai du réécrire le programme différemment) |
| Expérience personnelle |
| Un projet dûment mené qui m'a permis de consolider mon expérience en gestion de projet. Une certaine satisfaction en regardant le résultat final. |

6.1.3 Bilan individuel de Arno Ramm

| |
|--|
| Points positifs |
| Possibilité d'approfondir des langages déjà connus (Python, Flask) Travail sur la vulgarisation : explication de la base de données et des principes au groupe, des templates... |
| Difficultés rencontrées |
| Les seules difficultés rencontrées pour ma part auront été la répartition des tâches ainsi que l'organisation du groupe : beaucoup de tâches ne pouvaient pas être fragmentées entre plusieurs personnes, et chacun possédait son rythme de travail. |
| Expérience personnelle |
| Je garde un assez bon souvenir de ce projet, le groupe a su effectuer des réunions régulières et tout le monde était très réactif. Cela a été pour moi une bonne occasion d'approfondir des connaissances déjà acquises, même si certaines phases étaient un peu "frustrantes" car mes parties étaient terminées souvent avant celles des autres de par mon expérience précédente en informatique. |

6.2 Travail réalisé

Ce tableau comptabilise le nombre d'heures que chaque membre de l'équipe projet a consacrées aux différentes tâches. Les heures de réunions n'ont pas été incluses.

| Membres | Rachel | Alexandre | Arno |
|----------------------------------|--------|-----------|------|
| Travail de secrétariat | | | |
| Compte-rendus | - | - | 3h |
| Gestion de projet | 4h | 3h | - |
| Rédaction du rapport | 10h | 6h | 2h |
| Recherche d'informations | 6h | 4h | 3h |
| Partie conception | | | |
| Base de données : conception | - | 3h | 9h |
| Base de données : implémentation | 3h | 40h | 2h |
| Site internet : implémentation | - | 4h | 30h |
| Site internet : requêtes | 7h | 3h | 7h |
| Appréciation du travail | | | |
| Tests | 16h | - | - |
| Performance | - | 4h | - |
| Total | | | |
| | 46h | 67h | 66h |

TABLE 6.1 – Charge de travail

Bibliographie

- [1] ANONYME. *create-database*. URL : <http://www.w3big.com/fr/sqlite/sqlite-create-database.html>.
- [2] ANONYME. *Sqlite3*. URL : <http://python-simple.com/python-autres-modules-non-standards/sqlite3.php>.
- [3] CONCOURS CENTRALE-SUPELEC. *Rapport 2020 MP*. URL : <https://www.concours-centrale-supelec.fr/CentraleSupelec/2020/MP/rapCS2020MP.pdf>.
- [4] MAXICOURS.COM. *Contraintes d'intégrité*. URL : <https://www.maxicours.com/se/cours/les-contraintes-d-integrites/>.
- [5] WIKIPÉDIA. *Formes normales*. URL : [https://fr.wikipedia.org/wiki/Forme_normale_\(bases_de_donn%C3%A9es_relationnelles\)](https://fr.wikipedia.org/wiki/Forme_normale_(bases_de_donn%C3%A9es_relationnelles)).