Ministère de l’Enseignement Supérieur de la Recherche et de l’Innovation

Université Virtuelle du Sénégal

UFR Sciences et Technologies

Département Informatique

Intelligence Artificielle



**Rapport de projet**

Projet Scala

**Sujet**:

**Conception d’une application de gestion des emplois du temps d’un Etablissement scolaire**

Réalisé par : M. Rodrigue Eloi MANGA et Abdou DIOP

Professeur : Mr Djibril MBOUP

2020 - 2021

SOMMAIRE

[Introduction Général 3](#_Toc78622162)

[1 Présentation des outils utilisés 4](#_Toc78622163)

[1.1 Scala 4](#_Toc78622164)

[1.2 Intellij 5](#_Toc78622165)

[1.3 Base de données H2 5](#_Toc78622166)

[1.4 Scala Swing 6](#_Toc78622167)

[2.1 Identification des acteurs et des fonctionnalités du système 6](#_Toc78622168)

[2.2 Diagramme de Classe 6](#_Toc78622169)

[3 Guide d’utlisation de l’application 6](#_Toc78622170)

[Conclusion 11](#_Toc78622171)

## ***Introduction Général***

Ce projet entre dans le cadre de la formation en Intelligence Artificielle que nous suivons à travers l’UVS et qui nous permettra d’acquérir des connaissances et des compétences en analyse de données, en programmation, en machine learning, etc. c’est un projet de programmation fonctionnelle en Scala pour lequel il nous est demandé d’implémenter une application pour la gestion de l'emploi du temps des cours et des notes des étudiants d’un établissement. L’objectif de ce projet est de nous permettre de consolider et d’approfondir les concepts de programmation fonctionnelle avancée étudiés en classe, de maîtriser Scala et de développer de nouvelles compétences. Dans ce document, nous présenterons tout d’abord les outils utilisés pour le développement de l’application, des outils tels que Scala qui est un langage de programmation, l’environnement de développement intégré Intellij, la base de données H2 pour la gestion et le stockage des données de l’application et le module Scala\_Swing pour l’interface graphique. Puis modéliser l’application à travers le diagramme de classe. Nous présenterons également un guide d’utilisation détaillant toutes les différentes parties de l’application afin de pouvoir l’utiliser avec aisance et convivialité. Pour finir, nous donnerons une conclusion dans laquelle nous mettons en vue le savoir acquis durant la réalisation du projet ainsi que nos perspectives.

## ***1 Présentation des outils utilisés***

[1.1 Scala](#_Toc31015563)

Scala est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [multi-paradigme](https://fr.wikipedia.org/wiki/Paradigme_(programmation)) conçu à l'[École polytechnique fédérale de Lausanne](https://fr.wikipedia.org/wiki/%C3%89cole_polytechnique_f%C3%A9d%C3%A9rale_de_Lausanne) (EPFL) pour exprimer les modèles de programmation courants dans une forme concise et élégante. Son nom vient de l'anglais Scalable language qui signifie à peu près « langage adaptable ».

Scala intègre les paradigmes de [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) et de [programmation fonctionnelle](https://fr.wikipedia.org/wiki/Programmation_fonctionnelle), avec un [typage statique](https://fr.wikipedia.org/wiki/Typage_statique). Il concilie ainsi ces deux paradigmes habituellement opposés et offre au développeur la possibilité de choisir le paradigme le plus approprié à son problème. Il est prévu pour être compilé en [bytecode Java](https://fr.wikipedia.org/wiki/Bytecode_Java) (exécutable sur la [JVM](https://fr.wikipedia.org/wiki/Machine_virtuelle_Java)).

Si on souhaite l'utiliser exclusivement avec la JVM, il est alors possible d'utiliser les bibliothèques écrites en Java de façon complètement transparente. Ainsi, Scala bénéficie de la maturité et de la diversité des bibliothèques qui ont fait la force de Java depuis une dizaine d'années. De plus, il est possible d'invoquer du code écrit en Scala à partir de programmes écrits en Java ce qui facilite la transition de Java à Scala.

Scala peut également être combiné avec Spark pour assurer une certaine efficacité et performance en analyse et traitement des données.

Cependant Scala possède quelques inconvénients :

* Scala dispose d'un pool de développeurs et d’une documentation limités. .
* Difficile à apprendre. Syntaxiquement, Scala est très différent de Java traditionnel. Il présente un paradigme complètement différent.
* Manque de facilité d'adoption. Vous avez besoin d'une équipe qui est avide d'adopteurs pour que Scala soit faisable, vous trouverez rarement quelqu'un qui écrit uniquement en Scala.
* Rétrocompatibilité limitée. Chaque fois qu'une nouvelle version de Scala est publiée, elle est incompatible avec la version précédente, ce qui entraîne beaucoup de retard dans le produit.

[1.2 Intellij](#_Toc31015564)

Intellij IDEA également appelé « IntelliJ », « IDEA » est un environnement de développement intégré (en anglais [Integrated Development Environment](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9) - IDE) destiné au développement de logiciels informatiques reposant sur la technologie [Java](https://fr.wikipedia.org/wiki/Java_(technique)). Il est développé par [JetBrains](https://fr.wikipedia.org/wiki/JetBrains) (anciennement « [IntelliJ](https://fr.wikipedia.org/wiki/IntelliJ_(entreprise)) ») et disponible en deux versions, l'une communautaire, open source, sous [licence Apache 2](https://fr.wikipedia.org/wiki/Licence_Apache) et l'autre propriétaire, protégée par une [licence commerciale](https://fr.wikipedia.org/w/index.php?title=Licence_commerciale&action=edit&redlink=1).

Pour écrire et exécuter le code scala dans Intellij, nous avons utilisé l’outil SBT permettant la compilation du code Scala et de l'intégration avec de nombreux [frameworks de test](https://en.wikipedia.org/wiki/Test_automation) Scala. SBT prend en charge l’intégration avec l'interpréteur Scala pour une itération et un débogage rapides ainsi que l’intégration des projets mixtes Java/Scala.

[1.3 Base de données H2](#_Toc31015565)

H2 est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [relationnelles](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle) écrit en [Java](https://fr.wikipedia.org/wiki/Java_(langage)). Il peut être intégré à une application Java ou bien fonctionner en mode [client-serveur](https://fr.wikipedia.org/wiki/Client-serveur). Son fichier [jar](https://fr.wikipedia.org/wiki/JAR_(format_de_fichier)) est petit : environ 1 Mo[4](https://fr.wikipedia.org/wiki/H2_(base_de_donn%C3%A9es)#cite_note-4). C'est un logiciel [open source](https://fr.wikipedia.org/wiki/Open_source) distribué en licences [Mozilla Public License](https://fr.wikipedia.org/wiki/Mozilla_Public_License) ou [Eclipse Public License](https://fr.wikipedia.org/wiki/Eclipse_Public_License).

H2 supporte un sous-ensemble du standard [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language)[6](https://fr.wikipedia.org/wiki/H2_(base_de_donn%C3%A9es)#cite_note-6). Il propose des [interfaces de programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation)

 (APIs) [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language) et [JDBC](https://fr.wikipedia.org/wiki/Java_database_connectivity) ; toutefois il peut aussi utiliser le [pilote](https://fr.wikipedia.org/wiki/Pilote_informatique) [ODBC](https://fr.wikipedia.org/wiki/Open_Database_Connectivity) [PostgreSQL](https://fr.wikipedia.org/wiki/PostgreSQL) et se comporter en serveur PostgreSQL[7](https://fr.wikipedia.org/wiki/H2_(base_de_donn%C3%A9es)#cite_note-7). Les [tables](https://fr.wikipedia.org/wiki/Table_(base_de_donn%C3%A9es)) peuvent être créées en [mémoire vive](https://fr.wikipedia.org/wiki/M%C3%A9moire_vive) ou sur

[disque](https://fr.wikipedia.org/wiki/Disque_dur), elles peuvent être persistantes ou temporaires. Les [index](https://fr.wikipedia.org/wiki/Index_(base_de_donn%C3%A9es)) sont des [tables de hachage](https://fr.wikipedia.org/wiki/Table_de_hachage) et des [arbres](https://fr.wikipedia.org/wiki/Arbre_enracin%C3%A9) pour les tables en mémoire, et des [arbres B](https://fr.wikipedia.org/wiki/Arbre_B) pour les tables sur disque.

Pour utiliser H2 dans un projet, il faut l’ajouter dans les dépendances au niveau du fichier « build.sbt » à travers la ligne suivante :

"com.h2database" % "h2" % "1.4.190"

Le fichier application.conf doit également être configuré :

h2 ={  
 driver = "org.h2.Driver"  
 url = "jdbc:h2:file:./src/main/resources/emploisTemps;AUTO\_SERVER=TRUE"  
 connectionPool = disabled  
 keepAliveConnection = true  
 }

[1.4 Scala Swing](#_Toc31015566)

Il s'agit d'une bibliothèque d'interface utilisateur qui englobe la plupart de Java Swing pour Scala de manière simple. La hiérarchie des classes de widgets ressemble vaguement à celle de Java Swing. Le système d'événements scala-swing suit une approche différente de celle du système Java sous-jacent. Au lieu d'ajouter des écouteurs d'événement avec une interface particulière (telle que java.awt.ActionListener), une Reactorinstance annonce l'intérêt de recevoir des événements en appelant listenTo. Les éditeurs sont également des réacteurs et s'écoutent par défaut par commodité. Un réacteur contient un objet reactions qui sert d'endroit pratique pour enregistrer des observateurs en ajoutant des fonctions partielles qui correspondent à un modèle pour tout événement qui intéresse l'observateur.

A l’instar de H2, il faut également ajouter le module Scala\_swing dans les dépendances afin de pouvoir implémenter une interface graphique :

"org.scala-lang.modules" % "scala-swing\_2.13" %"2.1.1"

[***2 Modélisation de l’application***](#_Toc31015567)

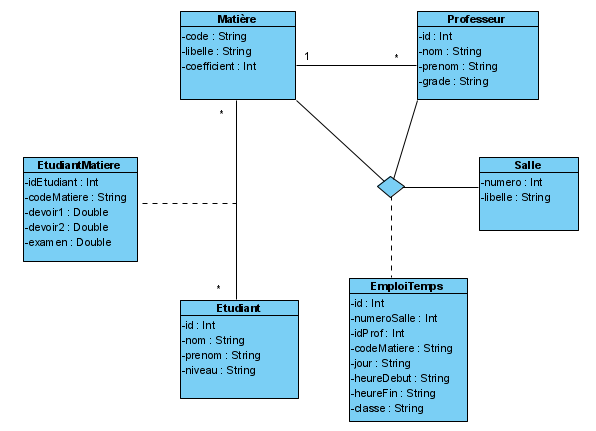
[2.1 Identification des acteurs et des fonctionnalités du système](#_Toc31015568)

Un acteur est une entité externe qui interagit avec le système, comme une personne humaine, un processus, … . Il agit sur le système (accès de lecture/écriture).

|  |  |  |
| --- | --- | --- |
| Fonctionnalités | Descriptions | Acteurs |
| Authentification |  | Tous les utilisateurs |
| Gestion des ressources | Pour l’ajout, la modification des salles, des matières, des élèves, et des enseignants | Administrateur  Proviseur |
| Générer automatiquement l’emploi du temps | Gérer l’emploi du temps | Administrateur, Proviseur |
| Gestion des utilisateurs | Ajout, suppression, modification des utilisateurs | Administrateur  Proviseur |
| Recherche de salles | Pour chercher des salles libres et les salles occupées | Administrateur |
| Allocation des salles | Pour attribuer des salles selon l’emploi du temps | Administrateur  Proviseur |
| Calcul des notes | Générer la moyenne de chaque élève et celle de chaque classe | Système |
| Calcul des statistiques | Permet à l’admin d’accéder aux statistiques de l’école | Système |

[2.2](#_Toc31015568) Diagramme de Classe

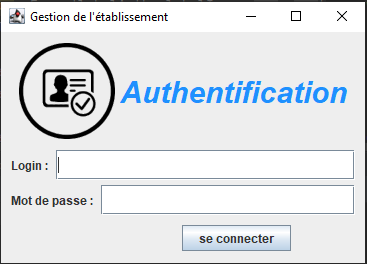
Le diagramme de classes est un schéma utilisé pour présenter les classes ainsi que les différentes relations entre celles-ci. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe. Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble. Les classes permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.



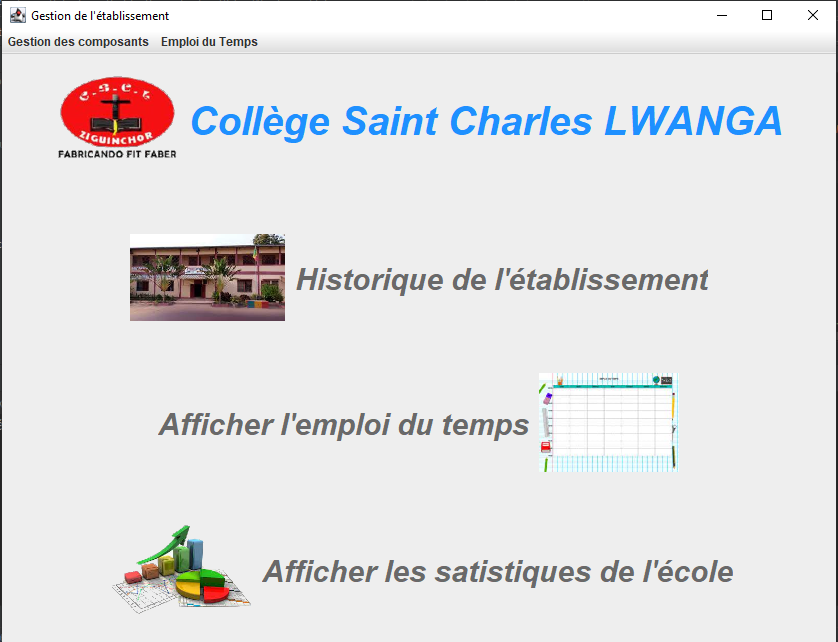
[3 Guide d’utlisation de l’application](#_Toc31015567)

L’application comprend plusieurs parties :

* Authentification : permet de restreindre l’accès à l’application, seuls les administrateurs peuvent se connecter avec leurs logins et mot de passe pour effectuer les modifications nécessaires. Dans notre cas, le login = admin1 et le mot de passe = passe123



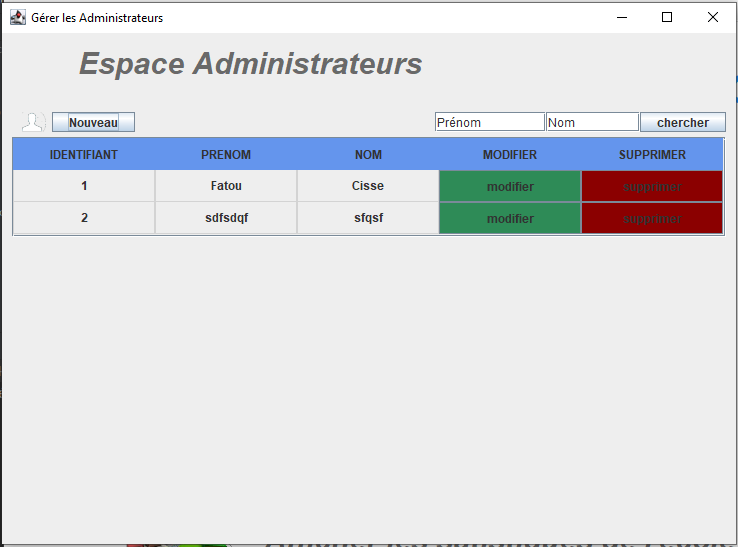
* Fenêtre principale : cette fenêtre contient une barre des tâches par laquelle l’utilisateur peut accéder aux autres compartiments de l’application. Cette barre des tâches comprend deux menus « Gestions des composants » et « Emploi du Temps » composés chacun de plusieurs sous-menus



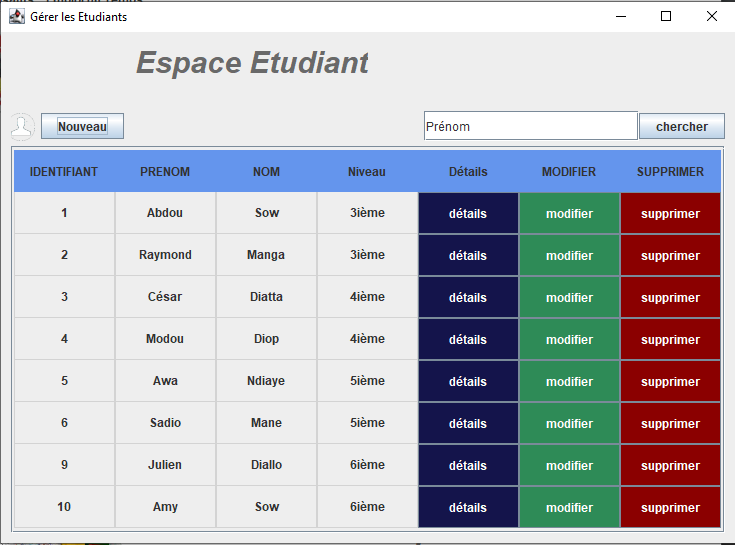
* Espace professeur : pour afficher et gérer (ajouter, modifier, supprimer) les informations des professeurs de l’école. Le Bouton « nouveau » sert à l’ajout d’un professeur, tant dis que pour modifier ou supprimer, il faut cliquer sur l’un des boutons correspondant aux données du professeur qu’on souhaite mettre à jour.

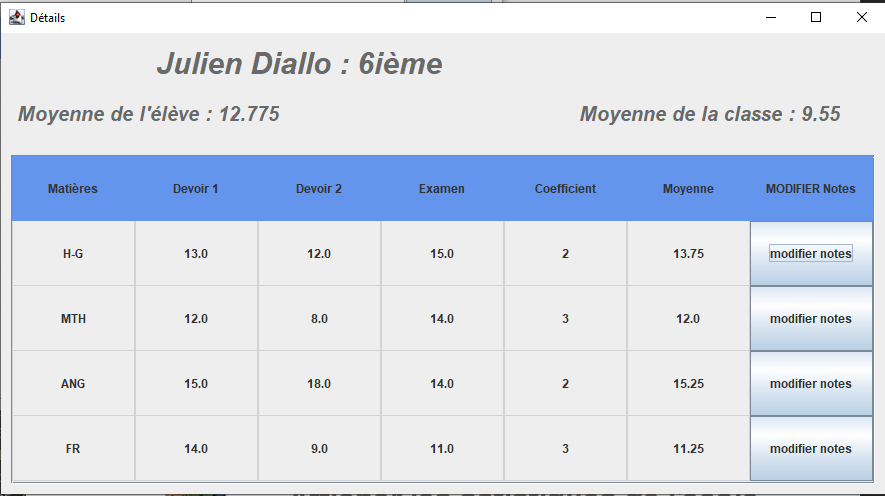


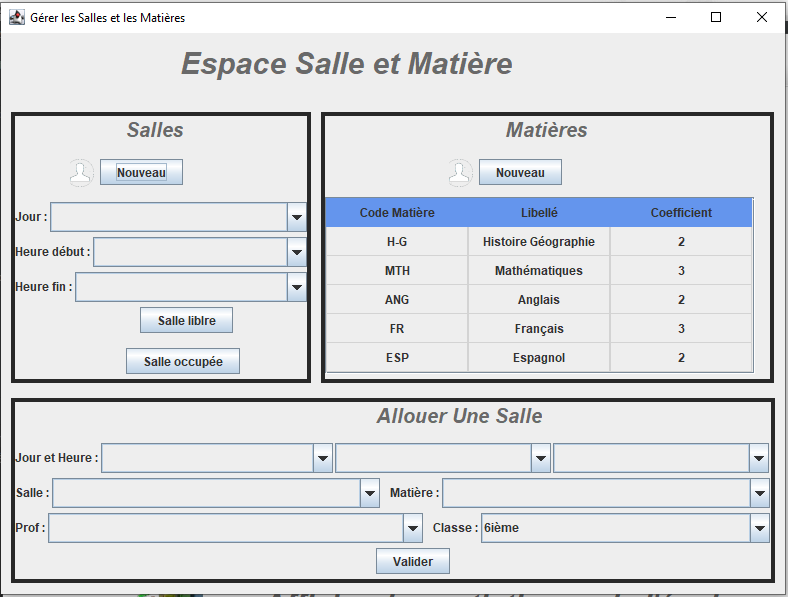
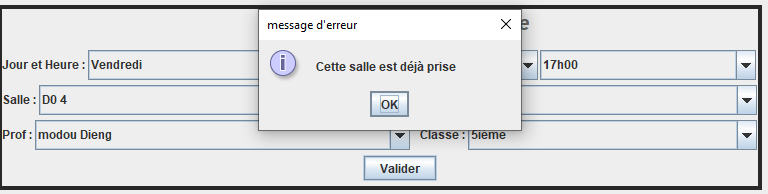
* Espace Admin : pour la gestion des différents administrateurs. Il est identique à celui des professeurs.

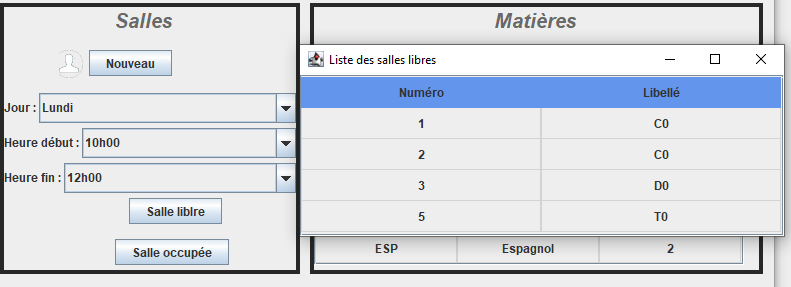


* Espace Etudiants : similaire à celui des Professeurs à la différence que l’utilisateur peut consulter les informations complètes des étudiants via le bouton « détails », c’est-à-dire les notes de devoir et examen, la moyenne de l’élève sur chaque matière ainsi que sa moyenne générale. Ces notes peuvent être modifiées en cas d’erreur.

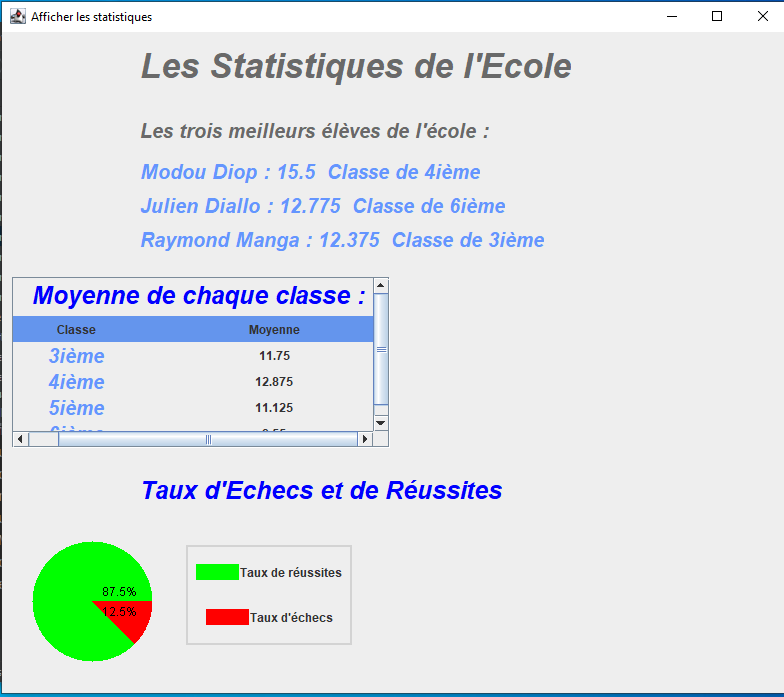




* Espace Salle Matière : cette fenêtre comprend trois parties, la partie gauche pour gérer les salles, afficher les salles libres entre deux intervalles de temps et les salles occupées. La partie droite pour les matières et la partie d’en-bas permettant d’allouer les salles selon leur disponibilité et par rapport à l’emploi du temps.



* Statistiques : ici, nous avons affiché les trois meilleurs élèves de l’établissement et représenter le taux d’échec et de réussite sous forme de diagramme circulaire



## ***Conclusion***

Ce projet consistait à concevoir une application de gestion de l’emploi du temps d’une école, ce que nous avons pu réaliser. A travers ce projet, nous avons mieux compris les notions de programmation fonctionnelle avancée et de programmation objet, maîtriser les différents outils de développement d’applications en Scala.

Cependant des difficultés ont été rencontrées pendant la phase d’implémentation de l’application. Des difficultés liées en général à la configuration de scala et sbt dans Intellij, et en particulier aux dépendances intégrées au niveau du fichier « build.sbt » correspondant à la base de données H2 et au module Scala\_swing.

Conscient que cette application n’est pas parfaite, nos perspectives seront d’apporter des améliorations afin d’assurer une bonne efficacité ainsi qu’une haute performance de l’outil final.