

PROJET ACADÉMIQUE TUTORÉ :

Analyse et conception d'une
application de gestion des
étudiants en JAVA

Membres du projet :

- ALAOUI ISMAILI Imaddedine
- BENCHEKROUN Othmane
- EL MOUDEN EL Mehdi
- EL ABD Othmane

Encadré par :

- BERRAJAA Achraf
- ZINEDDINE Mhamed

Année universitaire : 2020/2021

Remerciement

Dans un premier temps, nous tenons à remercier de tout cœur notre encadrant de projet Mr **BERRAJAA Achraf** pour avoir assuré le bon déroulement de notre projet avec succès et pour nous avoir légué ses conseils afin de rendre notre application java beaucoup plus flexible, scalable et optimal selon le cahier de charge suivi.

Dans un second temps, nous remercions également Mr **ZINEDDINE Mhamed** pour nous avoir épaulé à confirmer la véracité et l'exactitude de l'analyse en amont de notre projet à partir de la modélisation UML et des diagrammes qui caractérisent le design pattern de notre application.

Table des matières

Introduction.....	3
Contexte du projet (Définition du problème, interviews et solutions).....	3
Description du cahier de charge	4
Analysis and design of the application by using UML diagrams	5
Problems, opportunities, objectives and constraints matrix	5
Problem Statement and feasibility study	7
Responding to the system requirements by establishing Use Cases Models	9
Use Case Glossary	9
Use Case Narrative	12
Use Case Model Diagram	14
Data Modeling.....	15
Entity/Definition Matrix	15
Context Data Model	16
Key-Based Data Model	17
Fully Attributed Data Model.....	18
Object Oriented Analysis	19
Class Diagram.....	19
Data & Process Model CRUD Matrix (CREATE/READ/UPDATE/DELETE).....	20
Process Modeling.....	20
Event Decomposition Diagram.....	22
Event Diagrams.....	23
Context Diagram	25
Interaction and State Diagrams	26
Activity Diagram	26
Sequence Diagram.....	25
Administration de la base de données de l'application CRUD à partir de MySQL Workbench	28
MLD/MCD	28
Les différentes tables de la base de donnée.....	28
Implémentation et conception de l'application JavaFX	33
Logiciels et frameworks outillés.....	33
JavaFX	33
Feuilles de style en cascade (CSS)	34
Scene Builder	34
Le langage SQL	34
MySQL Workbench	35
JDBC (Java Database Connectivity).....	35
Microsoft Visio	35
Guide d'utilisation de l'application Java.....	36
Conclusion	43

Introduction

Notre projet académique, encadré par Mr BERRAJAA Achraf et Mr ZINEDDINE Mhamed, est centré autour de la création d'une application de gestion des étudiants scindée en deux interfaces d'authentification :

- Une interface professeur
- Une interface coordinateur

Afin de mettre en relief nos connaissances théoriques en programmation orientée objet (POO) et en système de gestion de bases de données (SGBD), ce projet-là nous a permis de nous familiariser avec le framework existant JavaFX en corrélation avec la base de données sur laquelle on agit tout en opérant des changements sur notre application.

Ces modifications apportées sont souvent liées en règle générale avec l'interfaçage de l'application : remplissage de la liste d'absence, indexation des coefficients de chaque matière et des notes de chaque étudiant, résultant donc à une décision du jury.

Les professeurs et les coordinateurs pourront alors s'authentifier afin de mettre en place leur propre réglementation sur ce qui leur est autorisé et affranchi de faire selon le cahier de charge que l'on explicitera ci-après.

Contexte du projet (Définition du problème, interviews et solutions)

La demande d'un système de gestion d'étudiants commence à se faire de plus en plus primordiale et impérative suite à la forte congestion que subit le système de gestion d'étudiants actuel au niveau des requêtes clients et serveur dont le dysfonctionnement et les dérèglements subsistent encore. De plus, les options configurables par les usagers ne sont pas optimales et propices. Ces principaux usagers touchés par la défaillance de ce système de gestion ne sont autre que les professeurs et les coordinateurs.

L'objectif de cette application est de manœuvrer un environnement de gestion d'étudiants pour les professeurs et les coordinateurs qui soit plus flexible, maintenable et convivial que l'ancien système en faisant borner la visibilité des options configurables selon le professeur de tel ou autre département afin d'éviter toute forme de discordance dans la liste des étudiants et des matières.

Description du cahier de charge

Il s'agit de faire immiscer et informatiser les plans d'étude instaurés par le service de scolarité dans la mesure où il est d'autant plus interactif et permissif pour les professeurs et les coordinateurs impliqués dans les départements et les filières des formations de l'université d'y apporter leurs champs de modification. A cette instance, l'application doit être en mesure de renflouer les fonctionnalités nécessaires au corps professoral et coordinateur.

Les professeurs doivent pouvoir :

- Renseigner la liste d'absence des bacheliers.
- Paramétrer et indexer la grille d'évaluation de leur propre module (coefficient de TP et des partiels, quantifier le nombre de séances selon sa nature : CM, TD, TP...).
- Renseigner les notes finales de chaque étudiant selon la session choisie (session normale ou session de rattrapage) et y émettre le verdict du jury de délibération (Validé, non validé, passage en session de rattrapage...), le tout sous un format PDF imprimable.
- Consulter l'emploi du temps d'affectation de chacun des cours ou TD prodigués.

Les coordinateurs doivent pouvoir :

- Procéder à l'ajout ou la suppression d'un étudiant de la liste des bacheliers (admission parallèle ou cas de démission ou d'ajournement définitif des élèves)
- Modifier les informations personnelles ou les informations d'état d'un bachelier
- Modifier les informations personnelles ou les informations d'état d'un professeur
- Consulter l'historique des modifications établies par le professeur

Analysis and design of the application by using UML diagrams

Problems, opportunities, objectives and constraints matrix

PROJECT :	University Technology Management System	PROJECT MANAGER :	Z. Mhamed
CREATED BY :	A. I. Imadeddine B. Othmane A. Othman M. Mehdi	LAST UPDATED BY :	A. I. Imad Eddine
DATE CREATED:	06/01/2021	DATE LAST UPDATED :	07/01/2021

CAUSE AND EFFECT ANALYSIS		SYSTEM IMPROVEMENT OBJECTIVES	
Problem or Opportunity	Causes and Effects	System Objective	System Constraint
1. The current system has security and reliability problems leaving the system vulnerable to cyber-attacks and abusive behavior.	1. Konosys that holds information is disorganized and is web hosted which makes it vulnerable web attacks. 2. Konosys is difficult to keep up-to-date because it has been developed with a 2011 obsolete version of HTML and C#. 3. Leads to wasted effort in the field, unbilled hours, and dissatisfied clients.	1. Create a Java application internet independent.	1. Java application must have adequate security.
2. The proposed system offers an easier way to calculate and view the student records and keeps track of the professor's activity for the coordinator.	1. The proposed system named ironically "KonoriSys" would be built with end-user in mind. 2. Would lead to time saving operations in term of attendance checking and student record management .	1. Create optimized algorithms for student report and history tracking system	1. Algorithms should be institution valid and certified. 2. If not online then need to replicate data between master and copy.

	3. Would allow the branch the coordinator to monitor and administer the activity of the system.		
--	---	--	--

CAUSE AND EFFECT ANALYSIS		SYSTEM IMPROVEMENT OBJECTIVES	
Problem or Opportunity	Causes and Effects	Problem or Opportunity	Causes and Effects
3. The proposed system offers a simple intuitive and secured way to approach the university management problem and will increase user satisfaction.	1. KonoriSys is an app service and not a Web dependent service and only uses Machine-to-server communication. 2. Would propose an easy and optimized way to fix the university management problem. 3. Would lead to a better customer satisfaction. 4. Would save time and though money.	1. Create a searchable database of configuration information. 2. System should be easy to use and maintain	1. If not online, should create savable and synchronized copy of database.
4. The current system isn't user- friendly and difficult to navigate in without prior knowledge which leads to time wasting and inefficacy.	1. Leads to wasted effort in course, unbilled hours, and dissatisfied clients. 2. Out-dated interface. 3. Opens useless sub tabs that increases user confusion, frustration and unsatisfaction.	1. Create a user-friendly interface	1. The System should still be functional over practical.

Problem Statement and feasibility study

Brief Statements of Problem, Opportunity , or Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. The current system has security and reliability problems leaving the system vulnerable to cyber-attacks and abusive behavior	3 months	High	\$15600 (avg. 3 attack/wk @ \$100/attack)	1	New Development
2. The proposed system offers an easier way to calculate and view the student records and keeps track of the professor's activity for the branch manager	3 months	High	\$2080 (2 hrs/wk x 52 wks/year x \$20/hr wage plus benefits)	2	New Development
3. The proposed system offers a simple and intuitive way to approach the university management problem and will increase user satisfaction	3 months	Medium	Unknown	3	New Development
4. The current system isn't user-friendly and difficult to navigate in without prior knowledge which leads to time wasting and inefficacy	3 months	Medium	\$2600 (avg. 2 hrs/wk @ \$50)	4	New Development

List of Requirements

ID	Requirement	Classification
001	The system should allow the coordinator to view the student's performance and report	Functional
002	The system should allow the professor and the coordinator to edit and view their assigned course configuration information.	Functional
003	The system must make it easy for the coordinators to edit the schedules and the course configuration system .	Non-functional
004	If system is not always online then need to replicate data between master database and copies.	Non-functional
005	The system should allow Professor to submit student's grades	Functional
006	The login parts of the system should not be online.	Non-functional
007	The database editing, update data viewing part of the system should be online.	Non-functional
008	The grade and course info part of the system should have adequate security.	Non-functional
009	The professor and the coordinators should be able to view Student information but only the coordinator would be able to edit them and add new student lists to the database.	Functional
010	The professor should be able to view but not edit their courses schedules	Functional
011	Only the professor of a specific course should be able to enter and edit the grades of the assigned course	Non-functional
012	The Professor should be able to register attendance of their assigned course	Functional
013	Only Professors should be able to enter attendance.	Non-functional
014	The coordinator should be able to enter new professors and edit their personal info as well as view and edit all the schedules.	Functional
015	The coordinator should be able to view the logs activity of the professors	Functional

Responding to the system requirements by establishing Use Cases Models

Use Case Glossary

Use-Case Glossary			
ID	Use-Case Name	Use-Case Description	Participating Actors and Roles
001	Enter Course Config Info	This use case describes the event of editing the configuration information of the course and saving the result to the database, a log history is created after saving the users choice. It will be used by both the professor and the branch manager	Professor Branch Manager
002	Edit Student Info	This use case describes the event of editing the Student's personal information's (Name, Address, Date of Birth, schedule...)	Branch Manager
003	Edit Student Grade	This use case describes the event of editing or entering the student's grade of a specific course	Professor
004	Enter New Student List	THIS USE CASE DESCRIBES THE EVENT OF ENTERING A LIST OF NEW STUDENTS TO THE DATABASE	Branch Manager
005	Enter Schedule Info	THIS USE CASE DESCRIBES THE EVENT OF A BRANCH MANAGER ENTERING THE SCHEDULE'S SUBJECT TO A CHOOSE COURSE LEVEL IN A BRANCH TO A SPECIFIC TIME.	Branch Manager

006	Enter Attendance	THIS USE CASE DESCRIBES THE EVENT OF A PROFESSOR ENTERING THE LIST OF ATTENDANCE RELATED TO THEIR COURSE AND BRANCH , THIS ACTION CAN ONLY BE VALID IF THE COURSE IS CURRENTLY RUNNING OR IS PASSED , THIS USE CASE ALSO INCLUDES THE OPTION OF EDITING THE LIST OF ATTENDANCE IF A STUDENT GIVES A JUSTIFICATION OF NOT ATTENDING THE COURSE.	Professor
007	Enter Professor Info	This use case describes the event of a branch manager entering new professors to the database and editing the information's about the professors in the database and assigning each professor with their respective course and branch	Branch Manager
008	View Branch Schedule	This use case describes the event of a branch manager viewing the schedule for the whole branch or for a whole level	Branch Manager
009	View Course Schedule	This use case describes the event of viewing the specific schedule for the course of the teacher that request to view it. It will be used by both the professor and the branch manager	Professor Branch Manager
010	View Student Info	This use case describes the event of viewing the student info (Name, Address, Date of birth, Schedule, Grades...). It will be used by both the professor and the branch manager	Professor Branch Manager
011	View Activity History	This use case describes the event of a branch manager viewing the Activity and action history of professors for a whole branch, level or course	Branch Manager
012	View student list	This use case describes the event of a branch manager or a professor viewing the list of students	Branch manager /professor
013	Login	THIS USE CASE DESCRIBES THE EVENT OF A BRANCH MANAGER OR PROFESSOR LOGGING INTO THE SYSTEM	Branch Manager professor

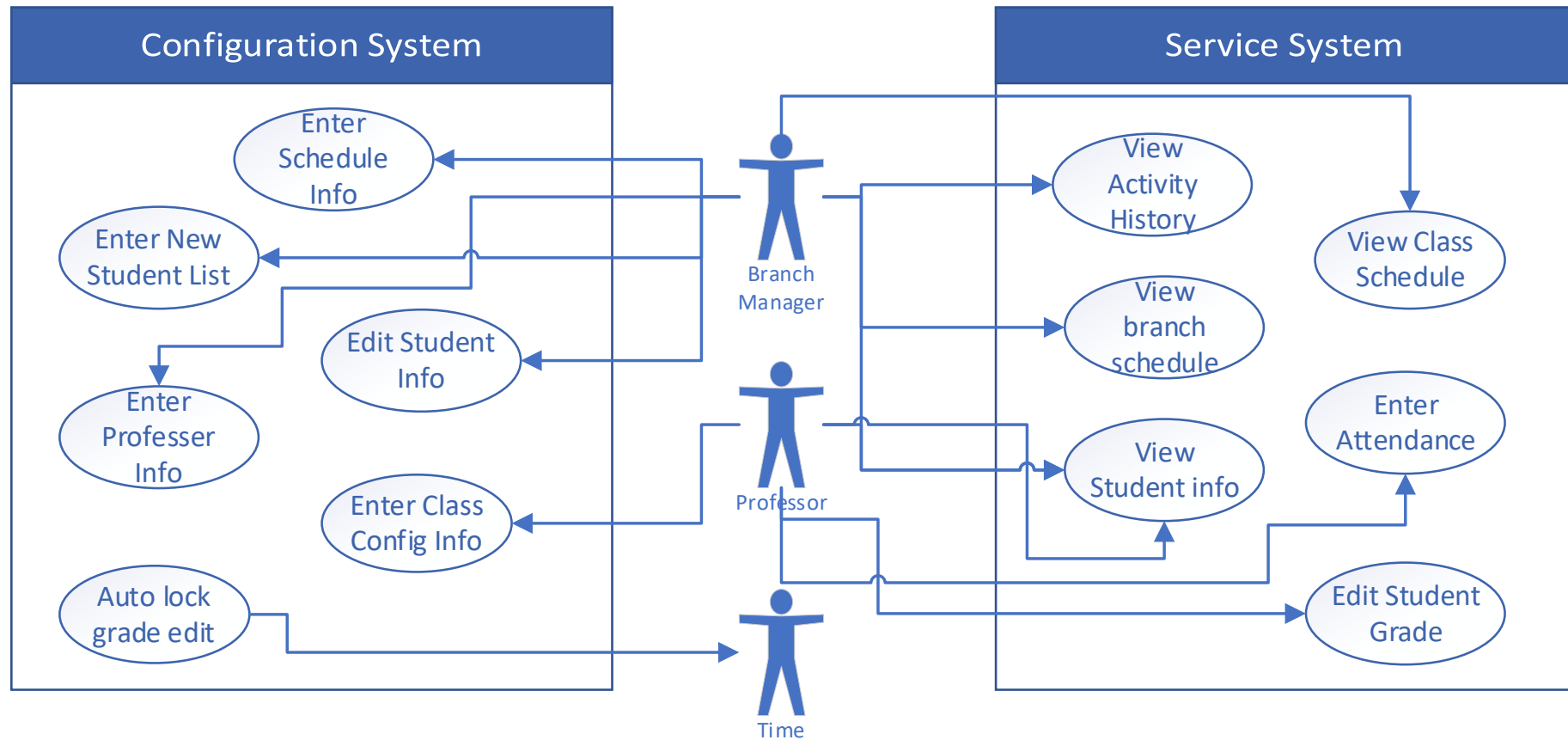
014	View attendance list	THIS USE CASE DESCRIBES THE EVENT OF VIEWING THE LIST OF STUDENTS WHO ATTENDED A COURSE, IT WILL BE USED BY THE BRANCH MANAGER OR THE PROFESSOR	Branch manager/ Professor
015	View course Info	THIS USE CASE DESCRIBES THE EVENT OF VIEWING THE INFORMATION OF THE SELECTED COURSE. IT IS USED BY THE BRANCH MANAGER AND THE PROFESSOR	Branch manager/ Professor

Use Case Narrative

UNIVERSITY TECHNOLOGY MANAGEMENT SYSTEM		
Author : A. I. Imad Eddine		Date : 17/02/2021
Use-Case Name :	Enter Course Config Info	Use Case Type : Business Requirement
Use-Case ID :	UTMS-001	
Priority :	Medium	
Source :	Requirement-002/003	
Primary System Actor :	Professor	
Other Participating Actors :	Branch Manager	
Other Interested Stakeholders:	None	
Description :	This use case describes the event of editing the configuration information of the course and saving the result to the database, a log history is created after saving the users choice. It will be used by both the professor and the branch manager.	
Precondition :	The user must have previously logged on so the system can identify the user as a particular professor or branch manager for a specific course.	
Trigger :	The use case is initiated when the user selects this option from the user interface.	
Typical Course of Events :	Actor Action	System Response
	Step 1: This use case is initiated when a user requests the option of editing or viewing	Step 2: The system responds by displaying a list of the course informations related to the user. Step 4: The system will display a confirmation button to save the changes.

	<p>information about the course.</p> <p>Step 3: The user may edit one of the informations about the course.</p> <p>Step 5: The user may confirm his choices.</p>	<p>Step 6: The system will verify that the user has entered appropriate data, sends the data to the database and leave a log of the performed tasks.</p>
Alternate Courses :	<p>Alt Step 2a: if there is no information about the course, the system will display an appropriate message and leave the editable text fields blank.</p> <p>Alt Step 5a: the user may ask to reset the informations to their original value in the database with an appropriate button.</p> <p>Alt Step 6a: if the data entered doesn't match the conditions set by the system will display an error message.</p>	
Conclusion :	This use case concludes when the user exits the Course Configuration Interface.	
Postcondition :	None	
Business Rules :	None	
Implementation Constraints and Specifications :	App programming to build an intuitive and easy to use the interface. Editing the database even if offline by keeping a trace of the changes and send them to the server when the app is connected again.	
Assumptions :	The configuration is a vital step for the calculation of the students grades and report	
Open Issues :	Need to determine whether or not when offline the program should create a copy of the database or save the database request to when online.	

Use Case Model Diagram



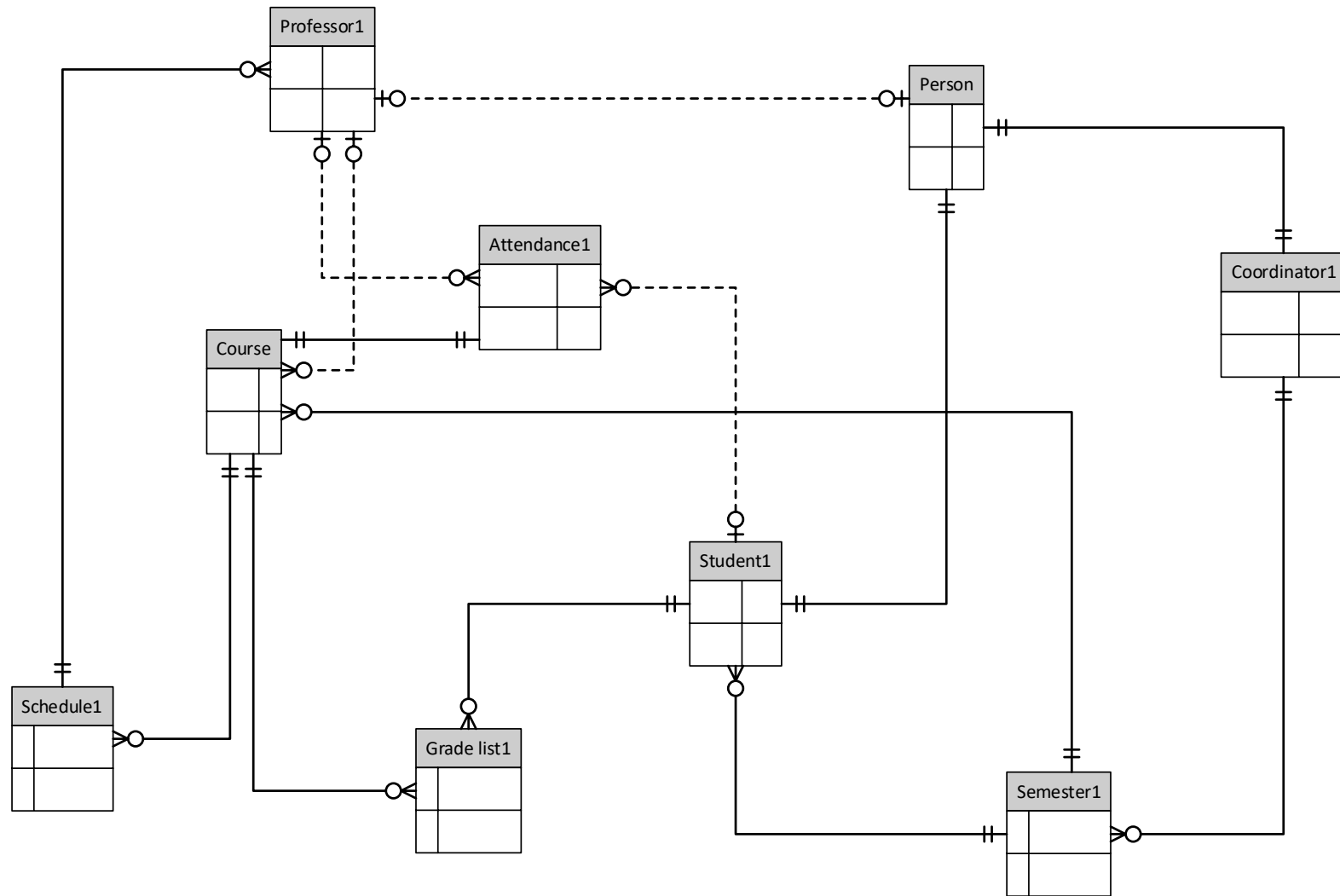
Use-Case Model Diagram

Data Modeling

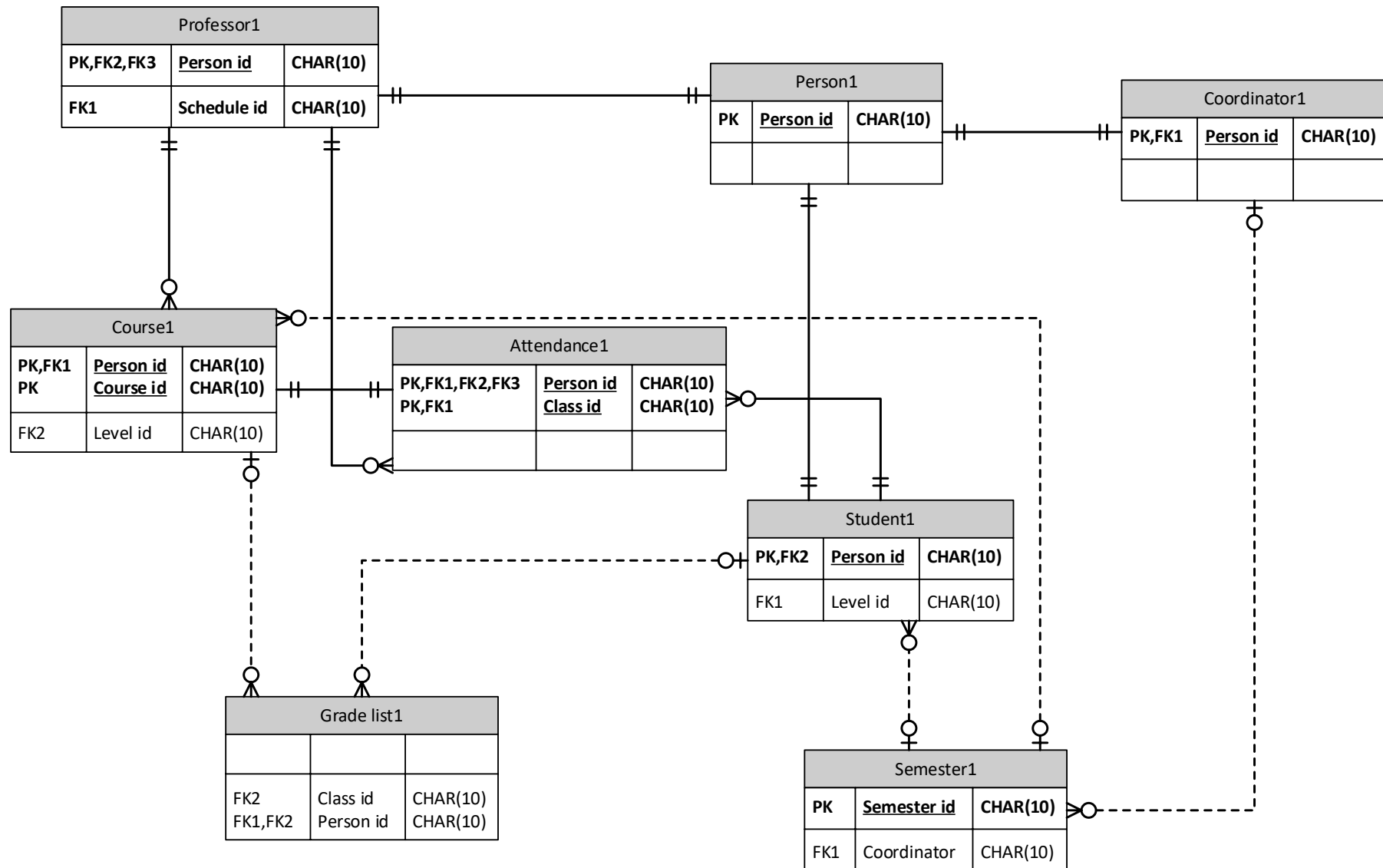
Entity/Definition Matrix

	BUSINESS DEFINITION
Major Entities	
Student	A business entity for whom work is done.
Grade list	An instance given to a Student
Schedule	A component that is viewed or edited
Attendance list	A component that keeps track of the Student attendance
Configuration	A piece of information concerning software configuration for the Student, Grade or Schedule and possibly for personnel
Professor	A business entity for whom work is done and achieved by.
Coordinator	An entity who administrates the system
Course	A component associated to business actors and system instances.
Optional Entities	
Level	A classification of Course, Student and professor
Branch	A classification of Course, Student and professor .

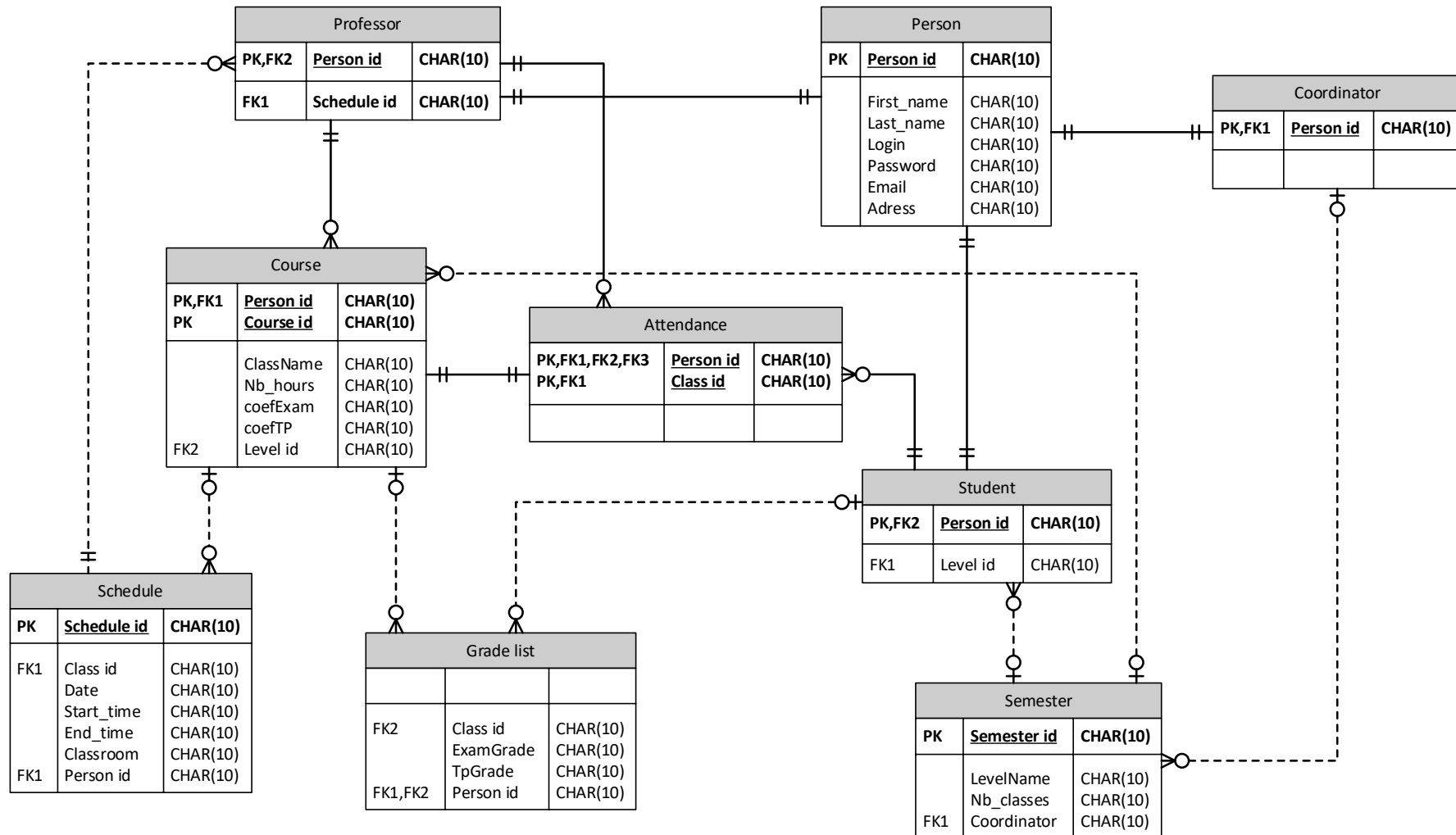
Context Data Model



Key-Based Data Model

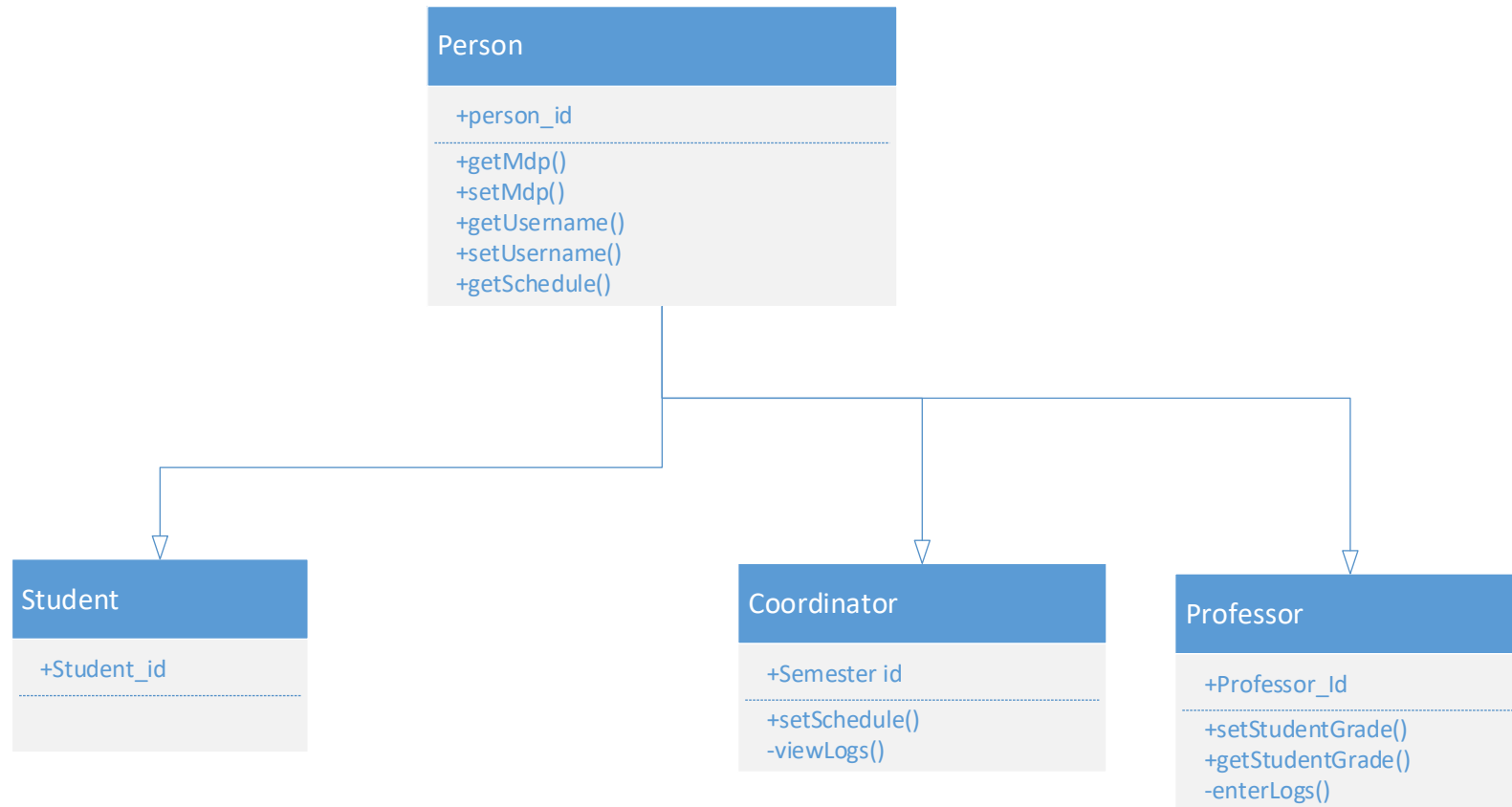


Fully Attributed Data Model



Object Oriented Analysis

Class Diagram



Data & Process Model CRUD Matrix (CREATE/READ/UPDATE/DELETE)

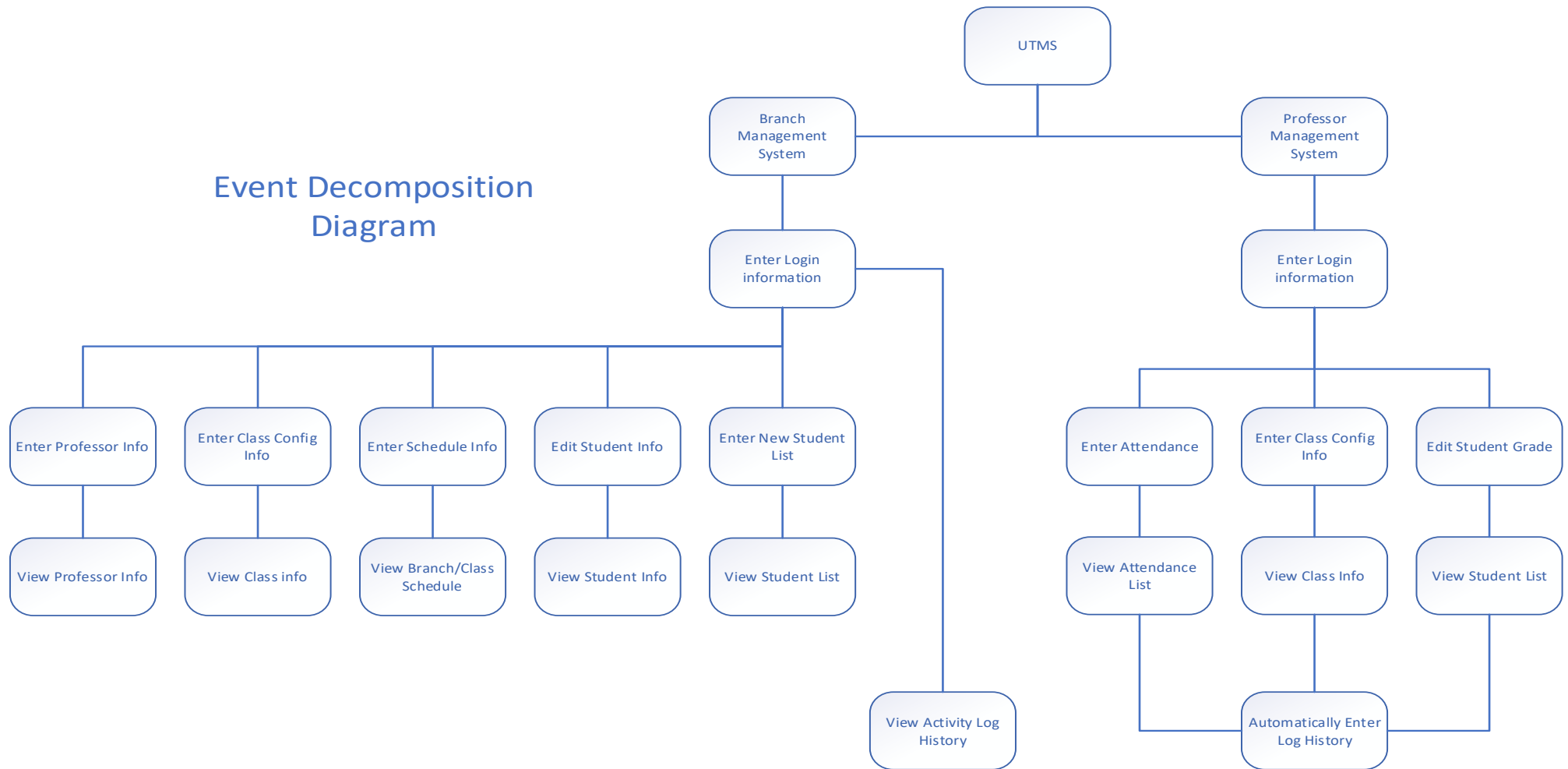
Entity/Attribute	Login	Enter Attendance	Enter Course Info	View Activity Log	Enter Student Grade	View Student List	View Student Info	Enter Schedule Info	Enter New Student list	Enter Professor Info	View Course Schedule	View Professor Info	View Course Info	View Attendance List	Auto Log Entry
Student		R			R	R	RU		CD					R	
Student id		R			R	R	RU		CD					R	
Level id		R			R	R	RU		CD					R	
Professor	R	R	R	R	R	R	R	R		C	R			R	R
Professor id	R	R	R	R	R	R	R	R		C	R			R	R
Coordinator			R	R		R	R	R	R	R	R	R	R	R	R
Coordinator id	R		R	R		R	R	R	R	R	R	R	R	R	R
Person									CRUD	CRUD					
First name							R		RU	RU		R			
Last name							R		RU	RU		R			
Login	R						R		RU	RU		R			
Password	R						R		RU	RU		R			
Email							R					R			
Adress							R					R			
Level															
Level id															
Attendance		U													
Grade					U		R						R		
Exam Grade					U		R						R		
TP Grade					U		R						R		

Course		R	RU												
Course id		R	RU		R					R			R	R	
Nb Courses		R	RU		R								R		
Coef Exam		R	RU		R								R		
Coef TP		R	RU		R								R		
Schedule															
Date		R						CRUD		R	R			R	R
Start time		R						CRUD		R	R			R	
End time		R						CRUD		R	R			R	

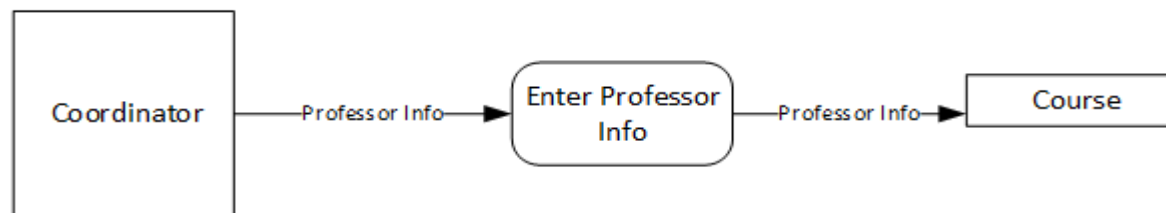
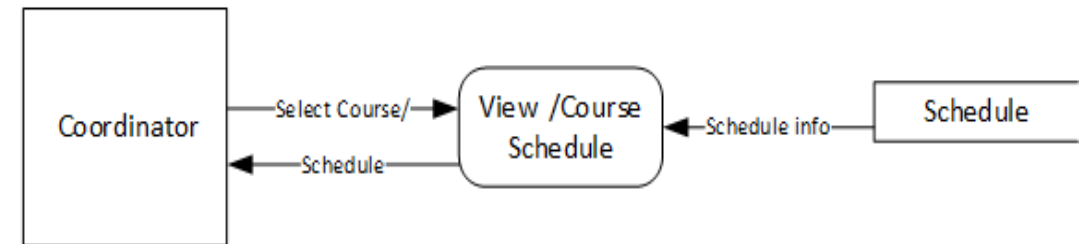
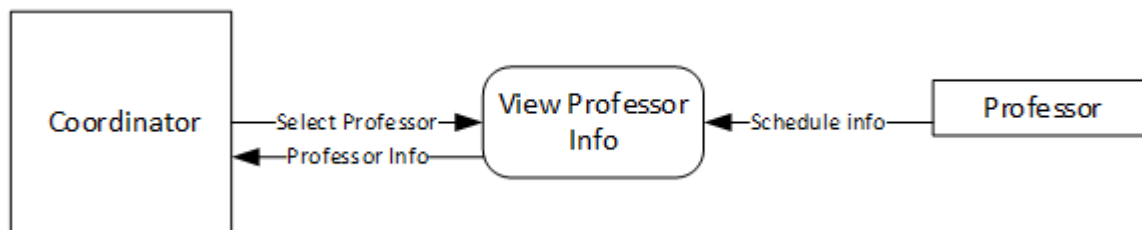
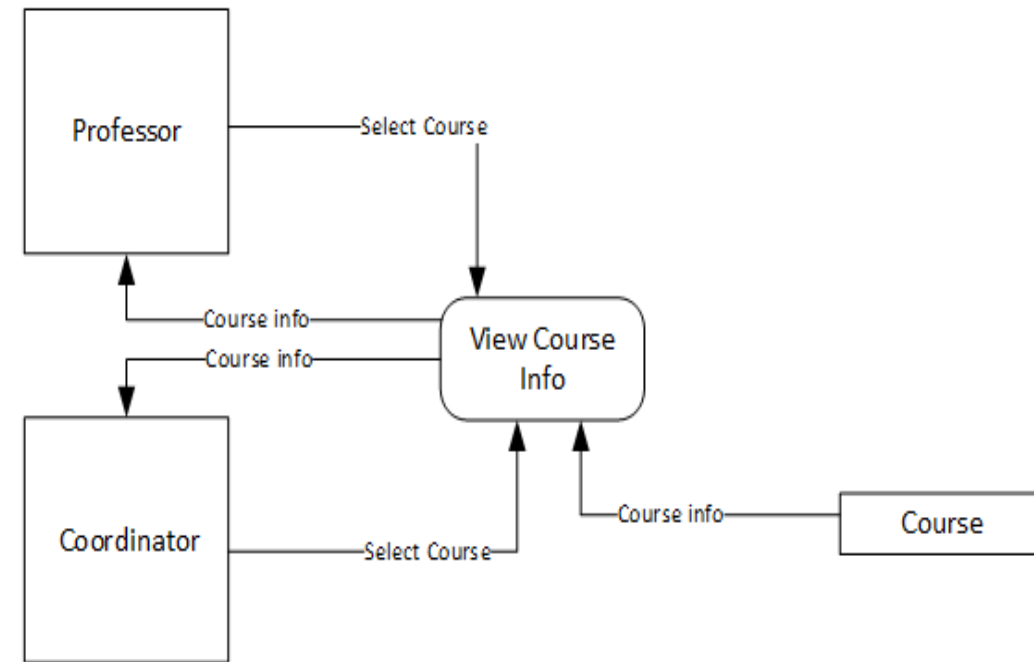
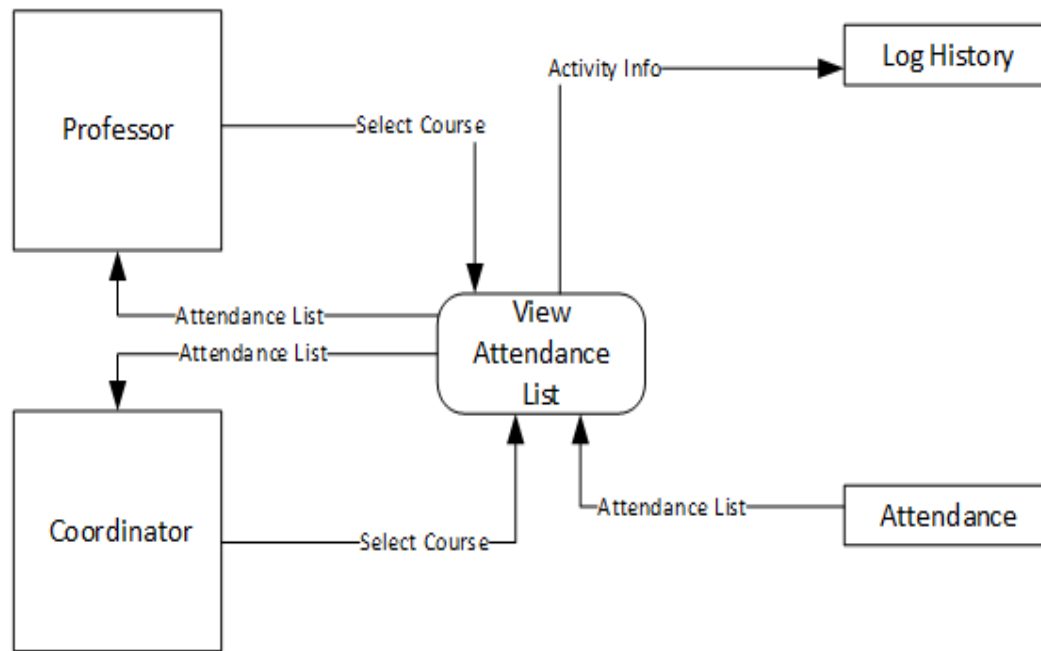
Process Modeling

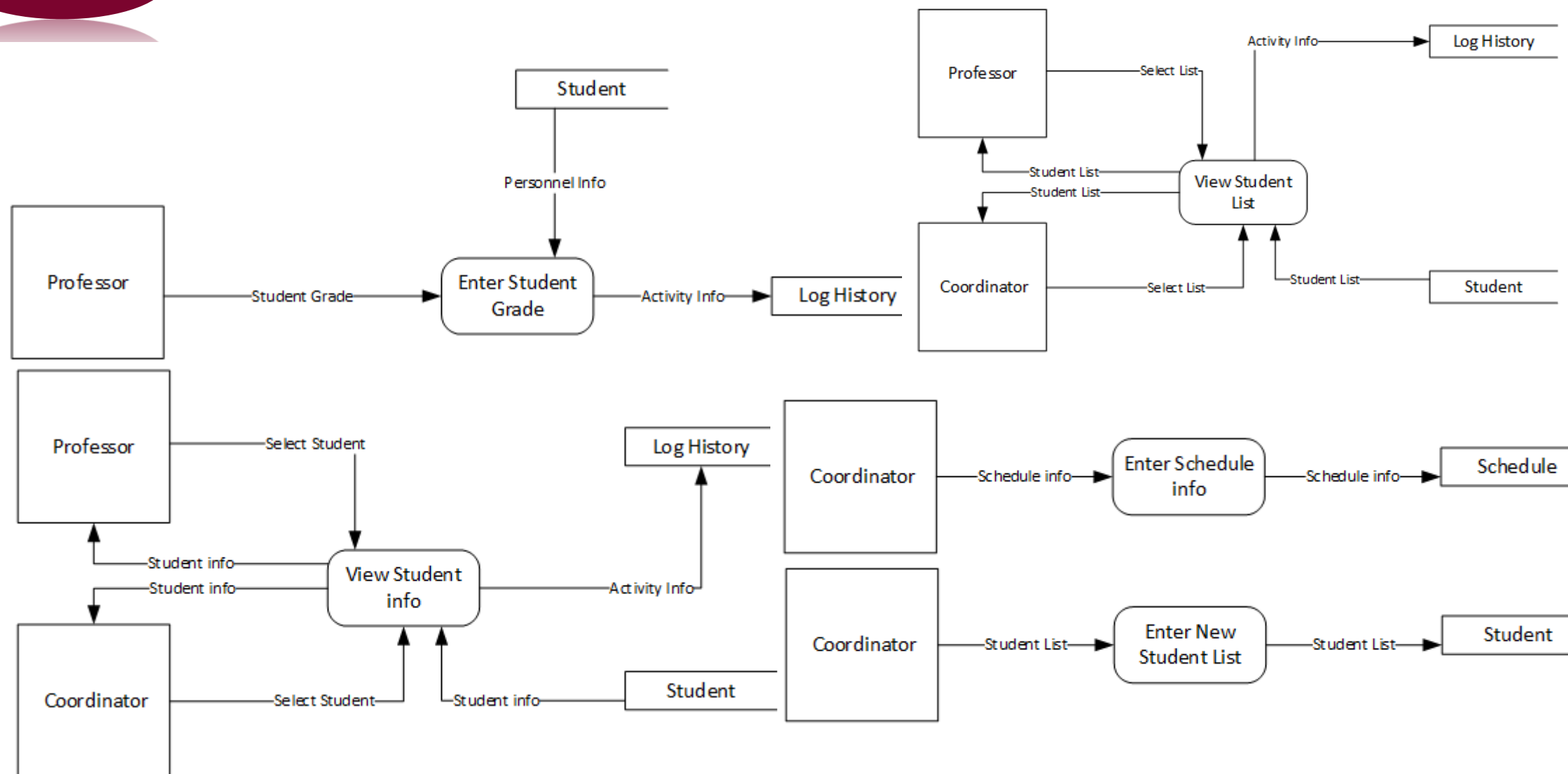
Event Decomposition Diagram

Event Decomposition
Diagram

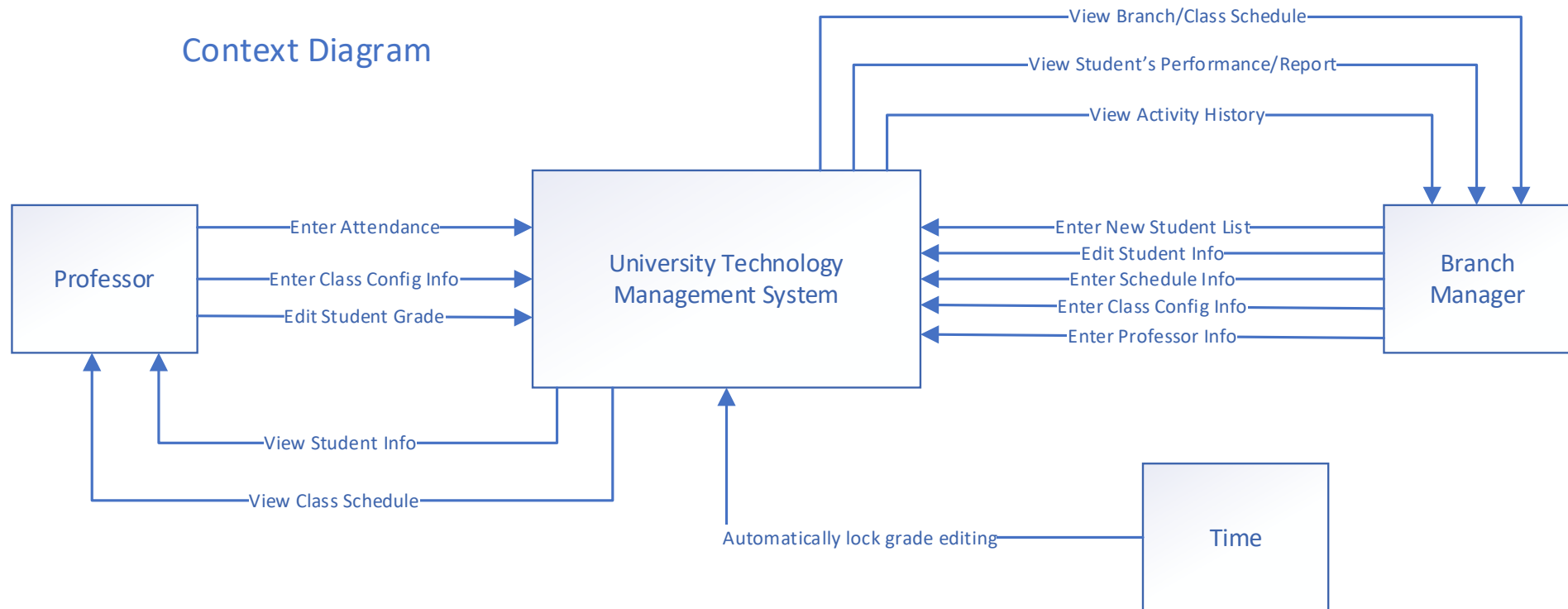


Event Diagrams (for each use case)



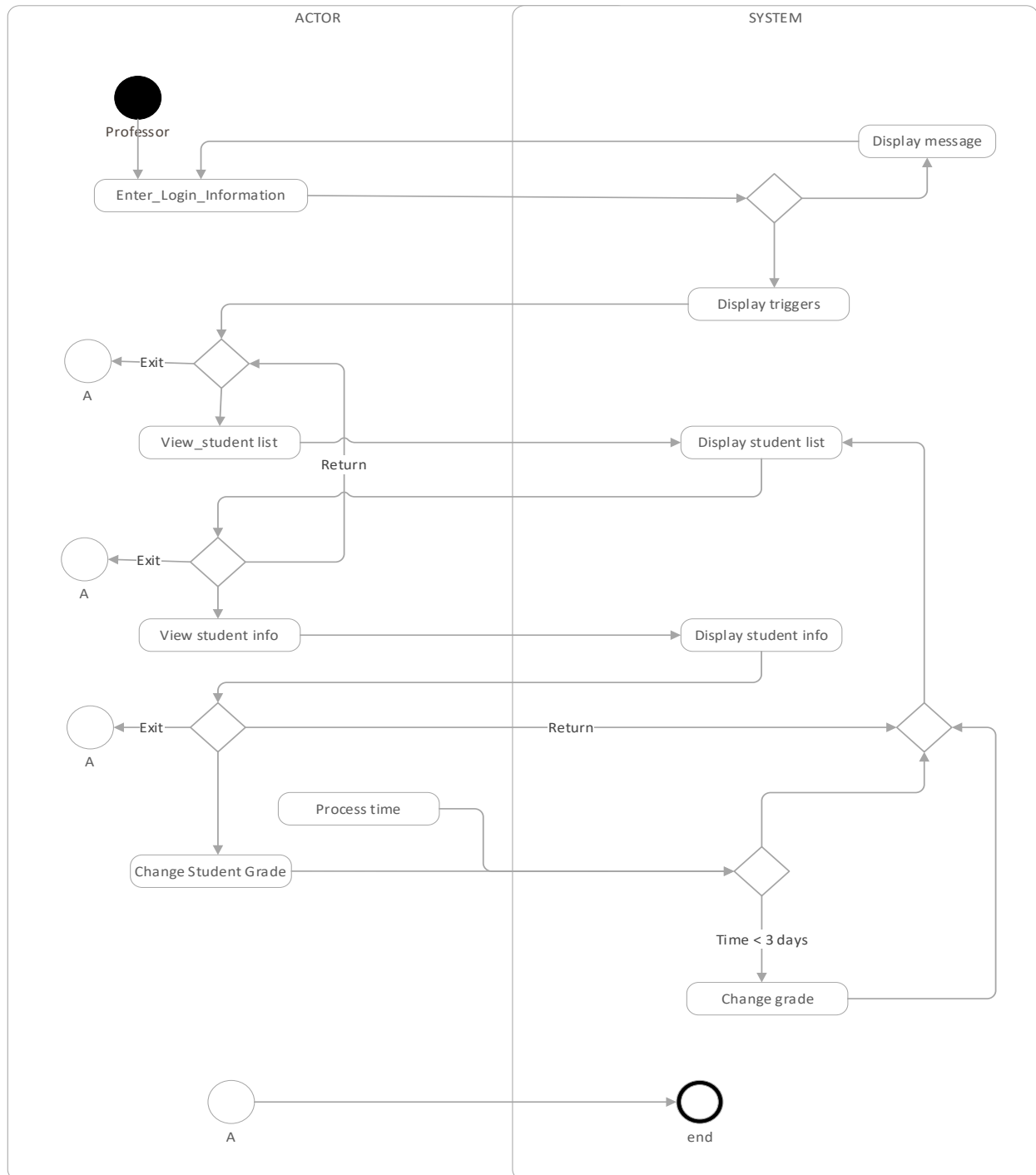


Context Diagram

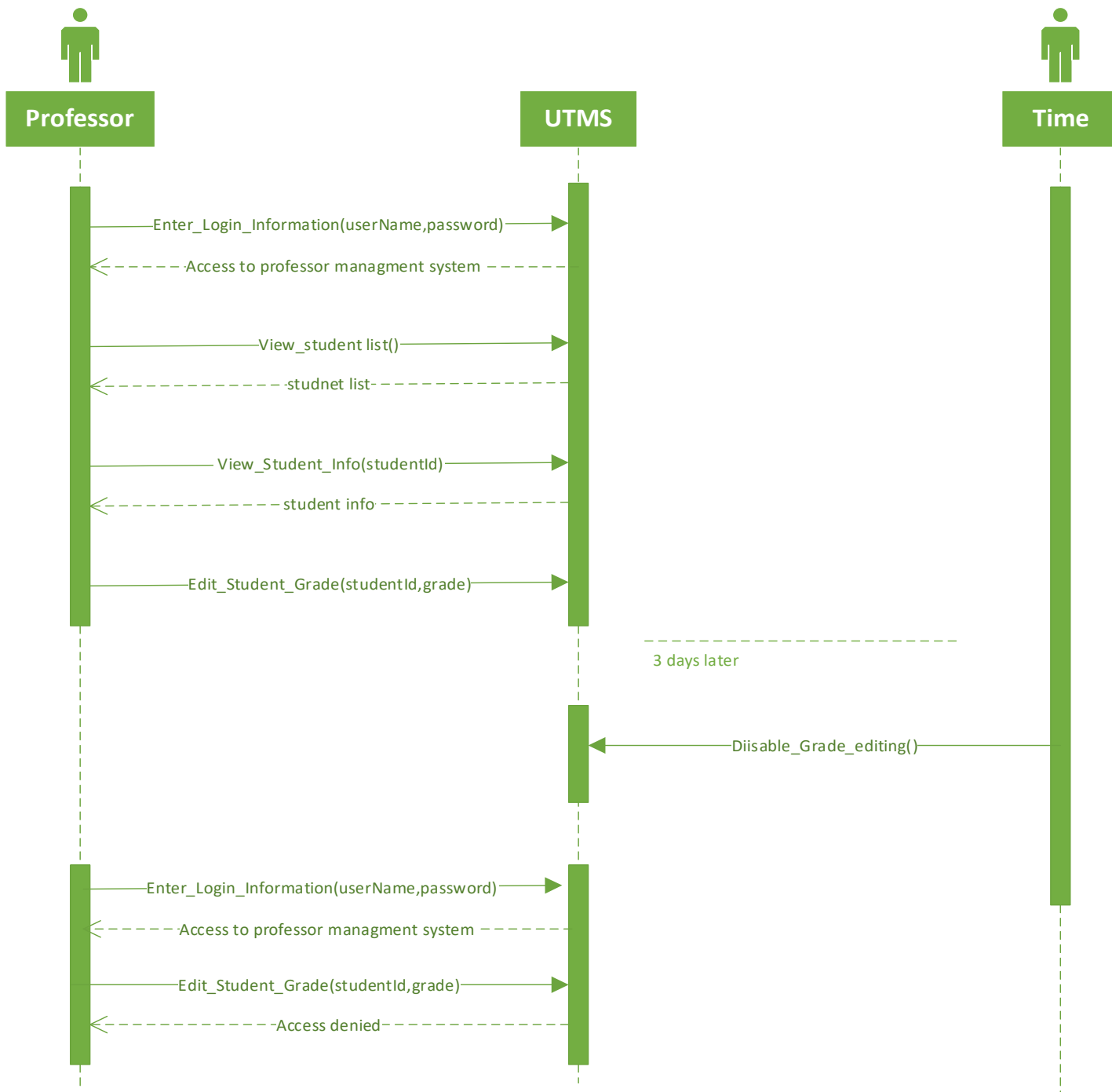


Interaction and State Diagrams

Activity Diagram



Sequence Diagram



Administration de la base de données de l'application CRUD à partir de MySQL Workbench

MLD/MCD

Le processus de conversion d'un MCD en MLD, c'est-à-dire en passant par Merise ou par le diagramme des classes pour découler vers le modèle entité-association, a été remanié dans la partie de modélisation UML de la page 16/18. (Voir dans la section Data Modeling -> Fully Attributed Data Model ainsi que la section Oriented-Object Analysis -> Class diagram).







Les différentes tables de la base de donnée

La table coordinateur

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idcoordonateur	VARCHAR(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
mdp	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
filiereCoo	VARCHAR(4)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'N/A'
nom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
prenom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	




Ces différents attributs figurent dans la table "coordinateur". Ils représentent conséquemment l'identité du coordinateur à savoir son nom et son prénom ainsi que son mot de passe d'authentification et sa filière dont il est le chef de département. Les valeurs de ses attributs sont de type VARCHAR ne dépassant pas conventionnellement 40 voir 45 caractères au niveau du nom et du prénom (chez certains pays anglo-saxons il s'agit de 30 caractères environ).

La table étudiant

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 cne	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 nom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 prenom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 mdp	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 filiereEtu	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 semestreEtu	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL








La table "étudiant" possède à peu près les mêmes attributs qu'un coordinateur ou un professeur sauf qu'il est identifié de façon unique par un CNE. Un étudiant peut être aussi caractérisé par sa filière d'étude à partir de la deuxième année (dans le cas où ce dernier n'a pas dépassé la première année, on lui affectera par défaut la STPI - Sciences et Techniques pour l'Ingénieur comme filière). Le semestre d'étude permet une bien meilleure gestion et répartition des cours suivis afin de mieux classer les profs vacataires et missionnaires qui enseignent dans une période saisonnière de l'année et par conséquent un élève suit un cours semestriel et non pas annuel. La matière "algorithmique et structures de données" par exemple ne peut être enseigné qu'en S3 et non pas en S4 également.

La table filière

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idfiliere	VARCHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 nbmodule	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 module	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL







Identifiable par un idfiliere, cette table regroupe le nombre de modules étudiées dans cette filière ainsi que les modules y compris ses éléments de modules.

La table matière

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idmatiere	VARCHAR(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 nomMatiere	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 coefEcrit	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
 coefTP	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
 nbSeanceTP	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 session	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 module	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL



Cette table contient tout ce dont une matière doit de mettre à disposition à l'étudiant dans l'interface : coefficient de TP, de partiel, nombre de séances, type de session (normale ou de rattrapage) ainsi que la nature de son module qui est en quelques sorte la clé étrangère par rapport à la table module.

La table module

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idModule	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 nomModule	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 volumeHoraireModule	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 coefficientModule	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
 semestre	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 filiere	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Même informations que pour la table matière à quelques différences. La table module a réciproquement comme clé étrangère l'identifiant de la table filière et la table semestre.

La table semestre

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idSemestre	VARCHAR(3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 nbModule	VARCHAR(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

La table professeur

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 idprofesseur	VARCHAR(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
🔗 nom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
🔗 prenom	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
🔗 Mdp	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Nous remarquons qu'il n'y a presque aucune différence notable entre la table professeur et coordinateur à l'exception des droits qu'ils leur étaient conférés dans l'application et des requêtes SQL implémentées (les trois composantes génériques CREATE UPDATE ET DELETE sont accordées au coordinateur, là où le professeur n'a droit qu'à la composante READ en SQL).

La table note

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 idEtudiant	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
🔑 idMatiere	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'N/A'
🔗 noteECRIT	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0.00'
🔗 noteTP	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0.00'
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

La table noteModule

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 idEtudiant	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
🔑 idMatiere	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'N/A'
🔗 noteECRIT	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0.00'
🔗 noteTP	DECIMAL(4,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0.00'
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

La table professeur-matière

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 idprof	VARCHAR(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
🔑 idmatiere	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Une table intermédiaire qui permet de balayer les cardinalités many-to-many comme règle de passage en MLD, elle y draine toutes les clés étrangères relatives aux clés primaires des deux tables professeur et matière.

La table noteSemestre

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
◊ idEtudiant	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◊ idSemestre	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◊ noteSemestre	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

La note du semestre est beaucoup plus complexe que ça, d'où le fait qu'on ne peut pas la limiter à remplir le rôle d'attribut mais plutôt d'entités. L'identifiant de l'étudiant et du dit semestre sont ses clés étrangères.

Implémentation et conception de l'application JavaFX

Logiciels et frameworks outillés JavaFX

JavaFX figure parmi l'un des frameworks les plus voluptueux et ergonomiques dans l'interfaçage proche de la programmation web en développement front-end. Cette bibliothèque java tire son épingle de ses prédécesseurs spirituels Java Swing et AWT, d'anciennes bibliothèques utilisées durant les années 90. Il imbrique plusieurs modules et classes importantes dans la réalisation d'une fenêtre appelée "stage" dont le contenu de ce "stage" se nomme "scène". Ces dites "scènes" sont animées par des gestionnaires de position comme à titre d'exemple les HBox, VBox, Stackpane, Anchorpane... JavaFX abrite un grand nombre de classes instanciables sous forme d'objets comme les labels, les boutons, les textfields, les combo-box, les check-box, les table-views et bien d'autres composants graphiques.



JFoenix

L'une des librairies qui se joignent avec l'API JavaFX pour des collections de classes de boutons et de textfields beaucoup plus modernes et embellies. La stylisation est proche du material design utilisé par Google.



CalendarFX

Du même acabit que JFoenix sauf qu'il permet de fournir des formes graphiques de calendrier. Son usage nous a fortement aidé pour mettre en place l'emploi du temps dans notre application.



Feuilles de style en cascade (CSS)

Les feuilles de style en cascade, généralement appelées CSS, permettent de styliser et structurer les documents web développés majoritairement en HTML5, XML voir même en FXML dans le cadre de notre projet.



Scene Builder

Développé par la société Gluon, Scene Builder est un builder qui permet d'imbriquer et d'agrégier plusieurs classes disponibles dans JavaFX dans une interface de conception graphique, générant automatiquement le code contrôleur du fichier java relatif à la scène construite. Elle fait intervenir le langage FXML qui repose essentiellement sur le même format que le XML. Ce logiciel nous a permis d'économiser un temps précieux pour la conception des interfaces dynamiques sans avoir recourt à leur implémentation directe en Java.



Le langage SQL

Il s'agit d'un langage informatique normalisé servant à exploiter des bases de données relationnelle. Il permet également de rechercher, d'ajouter, modifier ou encore supprimer des données et l'organisation des données. Son usage ne s'est limité qu'aux opérations CRUD élémentaires aux besoins du cahier de charge de notre projet.



MySQL Workbench

L'un des célèbres systèmes de gestion de base de données (SGBD) en constante concurrence avec PostgreSQL et PHPMyAdmin. Il permet de mettre en place des bases de données et de donner une liberté inconditionnelle à l'utilisateur pour y modifier et ajouter des enregistrements dans plusieurs champs. MySQL Workbench permet d'assembler automatiquement le MCD et le MLD des bases de données créées avec tous ses attributs, ses clés primaires et étrangères ainsi que les cardinalités entre chaque table. Ce SGBD fournit automatiquement les requêtes SQL pour procéder à l'exécution manuelle de celles-ci. L'utilisateur n'aura donc pas besoin d'implanter par le modèle physique (SQL en l'occurrence) toute sa base de données et ses tables.



JDBC (Java Database Connectivity)

La connexion de notre BDD a été effectuée en passant par la classe Driver Manager incluse dans l'API JDBC. Elle recèle de nombreuses méthodes de traitement de requêtes SQL et de leur exécution directement dans l'environnement Java afin de faire communiquer le programme Java avec la base de données mise en marche à partir d'un SGBD (celui qu'on utilise à l'heure actuelle n'est autre que MySQL Workbench).

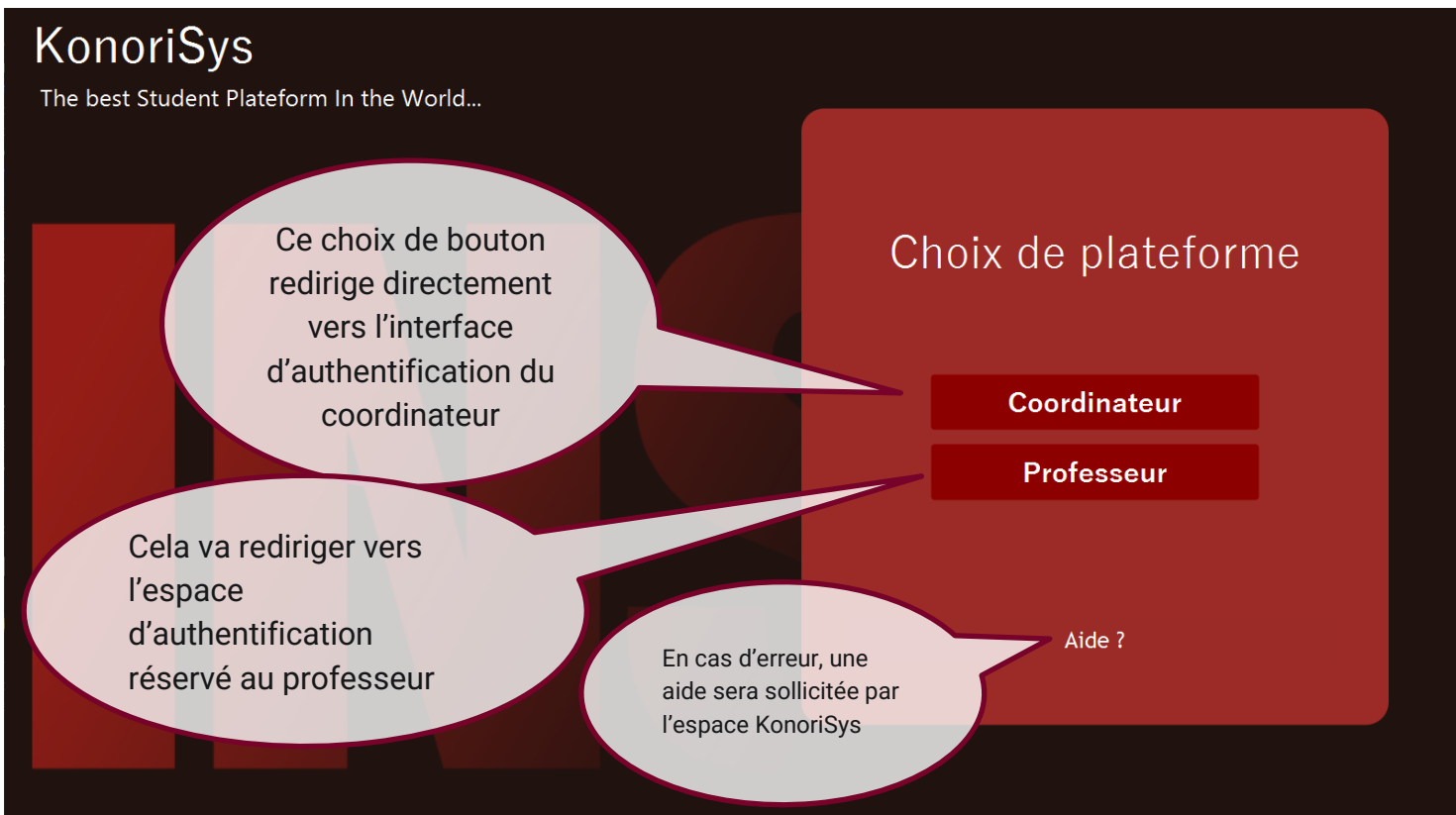
Microsoft Visio

Faisant partie de la famille de logiciels de Microsoft Office, Microsoft Visio est un outil qui façonne de toutes pièces toute sorte de diagramme partant des simples organigrammes au diagramme de classe, de cas d'utilisation, d'activité et bien d'autres utilisés en modélisation. Toute la partie analyse et modélisation a été élaborée à partir des diagrammes UML proposés par Visio.



Guide d'utilisation de l'application Java

Après avoir analysé l'ensemble des besoins du cahier de charge à partir des diagrammes UML, nous pouvons enfin passer au crible la partie phare de notre projet : l'implémentation de notre application physiquement à l'aide du package de notre projet Java. Ce package englobe et ramifie les contrôleurs des fichiers FXML, leur fichier main de chargement ainsi que les fichiers de l'accès à la base de données à partir des modules qui concernent l'API JDBC.



Deux cas de figure se présentent :

Espace Professeur

[< Retour](#)

Un champ de texte pour insérer l'identifiant du professeur

Un champ de texte pour insérer l'identifiant du professeur

Login

Connexion

[Mdp oublié?](#)

Espace Coordinateur

[< Retour](#)

Un champ de texte pour insérer l'identifiant du coordinateur

Un champ de texte crypté et masqué pour insérer le mot de passe du coordinateur

Login

Connexion

[Mdp oublié?](#)

Espace Professeur

< Retour

Login

Mdp ou identifiant Incorrect

identifiant faux

•••••

Connexion

Mdp oublié?

Un message d'erreur apparaît dans le cas où la base de données ne retrouve pas l'identifiant et le mot de passe insérés

KonoriSys

INSA
EURO-MEDITERRANEE

< Retour

Gestion Filière

Choix de Filière

SIC

Choix du Semestre

S3

Choix du Module

SIC 2

Choix du Matière

Algo Prog

Algo - Système d'exploitation

Valider

Si l'on se met sous la peau d'un enseignant du département SIC, le choix de filière semblera évident

Nom du module opté (SIC 2 est un initial qui renvoie à un ensemble de matières)

Nom de la matière choisie

Le choix de semestre si le professeur est vacataire

Aide

< Retour

Emploi du temps

Liste Etudiant

Saisie Notes Ecrits

Saisie Notes TP

Absence

Aide

Configuration de la Matière

Nombre de Scéance de TP

4

Coefficient Examen écrit

3

Coefficient Examen TP

1

Choisir la session

Session Normale ▼

Valider

Boutons qui renvoient respectivement :
les séances que le prof enseignera, la liste des étudiants de sa filière, ajout des notes et cocher la liste des élèves absents...

Session normale
ou session de rattrapage pour la
passation des partiels

< Retour

Emploi du temps

Liste Etudiant

Saisie Notes Ecrits

Saisie Notes TP

Affichage du jour
actuel dans le
calendrier

Aide

Emploi du temps

Calendars



Day

Week

Month

Year

Q Search

< Today >

mardi, 18. mai 2021



Agenda for 18 mai 2021 until 17 juin 2021

mardi	18 mai 2021
POO	12:00 to 13:00
mercredi	19 mai 2021
POO	12:00 to 13:00
jeudi	20 mai 2021
POO	12:00 to 13:00

All Day

14:00

15:00

16:00

17:00

18:00

19:00

20:00

21:00

22:00

23:00

23:56

Planification des
séances de POO
à venir
prochainement à
telle date, telle
heure

Emploi du temps

Calendars



Day

Week

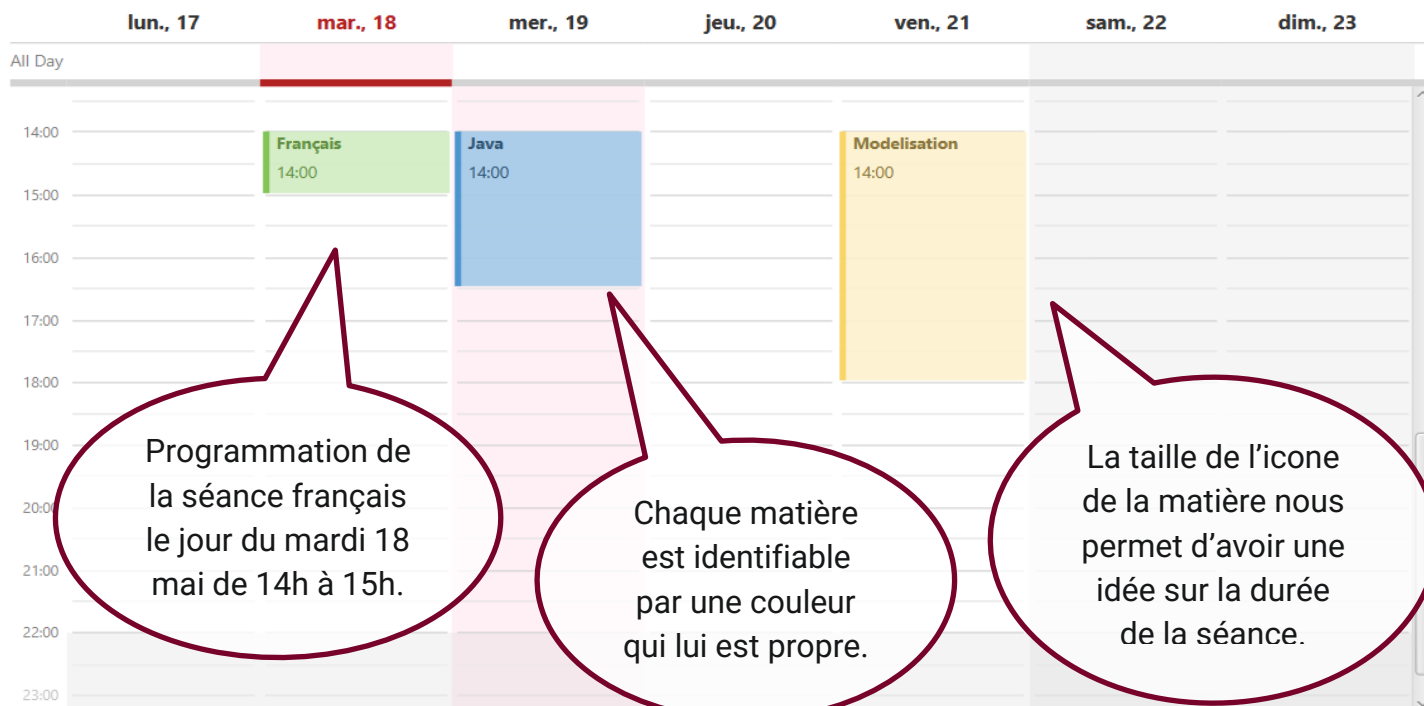
Month

Year

Q Search

< Today >

mai 2021



Programmation de
la séance français
le jour du mardi 18
mai de 14h à 15h.

Chaque matière
est identifiable
par une couleur
qui lui est propre.

La taille de l'icone
de la matière nous
permet d'avoir une
idée sur la durée
de la séance.

< Retour

Emploi du temps

Liste Etudiant

Saisie Notes Ecrits

Saisie Notes TP

La couleur verte
illustre la nature
de la séance :
CM, TD ou TP...

Aide

Emploi du temps

Calendars



Day

Week

Month

Year

Q Search

< Today >

2021

janvier

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
53	28	29	30	31	1	2	3
1	4	5	6	7	8	9	10
2	11	12	13	14	15	16	17
3	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31
5	1	2	3	4	5	6	7

février

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
5	1	2	3	4	5	6	7
6	8	9	10	11	12	13	14
7	15	16	17	18	19	20	21
8	22	23	24	25	26	27	28
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14

mars

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14
11	15	16	17	18	19	20	21
12	22	23	24	25	26	27	28
13	29	30	31	1	2	3	4
14	5	6	7	8	9	10	11

avril

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
13	29	30	31	1	2	3	4
14	5	6	7	8	9	10	11
15	12	13	14	15	16	17	18
16	19	20	21	22	23	24	25
17	26	27	28	29	30	1	2
18	3	4	5	6	7	8	9

mai

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
17	26	27	28	29	30	1	2
18	3	4	5	6	7	8	9
19	10	11	12	13	14	15	16
20	17	18	19	20	21	22	23
21	24	25	26	27	28	29	30
22	31	1	2	3	4	5	6

juin

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
22	31	1	2	3	4	5	6
23	7	8	9	10	11	12	13
24	14	15	16	17	18	19	20
25	21	22	23	24	25	26	27
26	28	29	30	1	2	3	4
27	5	6	7	8	9	10	11

juillet

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	28	29	30	1	2	3	4
27	5	6	7	8	9	10	11
28	12	13	14	15	16	17	18
29	19	20	21	22	23	24	25
30	26	27	28	29	30	31	1
31	2	3	4	5	6	7	8

août

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
30	26	27	28	29	30	31	1
31	2	3	4	5	6	7	8
32	9	10	11	12	13	14	15
33	16	17	18	19	20	21	22
34	23	24	25	26	27	28	29
35	30	31	1	2	3	4	5

septembre

octobre

novembre

décembre

KonoriSys

Print

17 mai 2021 to 23 mai 2021

mai 2021 juin 2021

	lun., 17	mar., 18	mer., 19	jeu., 20	ven., 21	sam., 22	dim., 23
All Day							
0800							
0900							
1000							
1100							
1200							
1300							
1400							
1500							
1600							
1700							
1800							
1900							
2000							

Zoom

< 1/1 >

Paper

View

Week

Paper

A4

Margin

Default

Time Range

Start

This Week

End

After

1

week

1 week will be printed

Calendars

Default

☒ Default

Timetable

☒ CM☒ TD

Options

☒ All Day Entries☒ Mini Calendar☒ Calendar Keys☐ Column Layout

Cancel

Continue

L'impression de
l'emploi du temps
est possible !

[< Retour](#)

Afficher la liste

Chercher un Etudiant

Saisie Notes Ecrits

Saisie Notes TP

En appuyant sur cette rubrique, le professeur peut renseigner la note de partiel de chaque étudiant.

Saisie Note Ecrit

Nom :

Prénom :

CNE :

Filière :

Note écrit :

Afficher

< Précédent

Suivant >

Valider

Passer à l'étudiant suivant pour y entrer sa note écrite.

[< Retour](#)

Emploi du temps

Liste Etudiant

Saisie Notes Ecrits

Saisie Notes TP

Aide

Liste Absence

Informations Enregistrés

Confirmer

Absence	StudentId	Nom	Prenom
<input type="checkbox"/>	1	Alaoui Ismaili	Imadeddine
<input type="checkbox"/>	2	Benchkroun	Othmane
<input type="checkbox"/>	3	Zaim	Othmane
<input type="checkbox"/>	4	Hamid	Louchkou
<input type="checkbox"/>	6	Bakati	Mohamed
<input type="checkbox"/>	7	Taxista	Mehdi
<input checked="" type="checkbox"/>	8	Statstsatsfit	Samir
<input type="checkbox"/>	9	SSSekkkoumi	Samir
<input type="checkbox"/>	10	Taha	Nabil
<input type="checkbox"/>	11	Idriss	Naoufe
<input type="checkbox"/>	12	Se9at	Hakima
<input type="checkbox"/>	13	Zaim	Adam
<input type="checkbox"/>	14	BOhi	Mohamed
<input checked="" type="checkbox"/>	15	Emond	Michael

Une recherche séquentielle pour afficher l'étudiant en question soit par son nom ou son prénom

Nous pouvons cocher la case d'absence si l'étudiant est jugé absent ou indisponible durant la séance.

INSA
EURO-MEDITERRANEE

< Retour

Passer à une autre fenêtre d'ajout d'un autre étudiant (Attention à bien valider les informations renseignées supra au risque d'y perdre son enregistrement)

Chercher un Etudiant

Saisie Notes

Saisie Notes TP

Aide

Ajout d'étudiant

Nom : Tastift

Prénom : Samir

CNE : 9

Filière : SIC

Année d'étude : 2

Ajouter un autre étudiant

Valider

Confirmation des informations de l'étudiant qu'on souhaite intégrer dans la liste des bacheliers

Conclusion

Après une multitude d'interviews qui nous ont permis de dépister les besoins du cahier de charge de l'application, toutes les fonctionnalités sont opérationnelles et l'ergonomie a été retravaillée et affinée afin de laisser beaucoup de liberté et de latitude aux professeurs et aux coordinateurs.

Nous sommes alors enthousiastes d'avoir érigé un tel projet de développement web de taille et d'envergure et d'y avoir instillé notre touche de créativité en raccord avec l'analyse de bout en bout du système de notre application avec maîtrise, perspicacité et parcimonie. Cela nous a tout autant positionné vers une bonne cohésion d'équipe et une synergie sans pareille dont chacun d'entre nous a pu parfaire et égrener des connaissances plus pointues et aguerries en SGBD relationnelle, en programmation orientée objet notamment au niveau des notions d'héritage, d'abstraction et d'interface.

Enfin, nous espérons avoir été au bout de notre tâche avec efficacité et savoir-faire et que notre application réponde en guise de solution qui aboutisse et dérape vers ce qui a été décrété par le plan d'étude du service de scolarité.