**Introduction générale**

L’informatique joue un rôle important dans le développement de l’entreprise et d’autre établissement. Elle intervient dans tous les domaines et tous secteurs pour les rendre plus faciles, plus précis et surtout bien gérés.

L’Institut Supérieur d’Informatique du Kef a été créé selon le décret n°06-1587 du 6 Juin 2006 portant création d’établissements d’enseignement supérieur et de recherche. En effet, l’ISI Kef est l’une des établissements universitaires de l’université de Jendouba.

Pour la réalisation de notre plateforme, on va visiter l’institut, situé au rue Salih Ayech Kef. Le directeur de cet institut nous a fourni toutes les informations nécessaires et suffisantes pour que nous atteindrons notre objectif afin de mettre en place une solution qui permet la gestion des activités administratif pour l’institut et faciliter le processus de connaitre les services fournis et de faire les inscriptions.

Le présent rapport a pour objectif de donner une idée claire sur le travail effectué pendant la réalisation du projet. Il s’articule autour de cinq chapitres relatifs aux différentes étapes de spécification, conception et développement d’application : Notre premier chapitre consacré à la présentation globale du projet, il est composé de quatre parties ; la première partie est réservée à la présentation de l’organisme d’accueil, la deuxième partie est réservée à la présentation du contexte du projet, la troisième partie est consacré à l’étude de méthodologie du travail et dans la dernière partie nous avons présenté le langage de modélisation adopté. Puis le chapitre deux est consacré à la spécification des besoins, la présentation de l’étude technique le diagramme de cas d’utilisation global, le diagramme de classe globale ainsi que le Backlog du produit. Les trois chapitres qui suivent sont consacrés pour la spécification, la conception et la réalisation des différents sprints. Enfin, le rapport se termine avec une conclusion générale et des perspectives qui peuvent servir à des éventuelles améliorations de plateforme.

Présentation du cadre de projet

# Introduction :

Dans ce chapitre, nous allons présenter le contexte général de notre projet. D’abord, nous allons exposer l’environnement du stage par une présentation de l’Institut Supérieur de l’informatique du Kef (ISI KEF). Ensuite, nous mettrons l’accent sur l’analyse et la critique de l’existant. Nous finirons par la solution proposée et par le modèle de développement adopté de notre projet.

**1. présentation du cadre de projet**

**1.1. Présentation de l’ISI**

L’Institut Supérieur d’Informatique du Kef a été créé selon le décret n°06-1587 du 6

Juin 2006 portant création d’établissements d’enseignement supérieur et de recherche.

En effet, l’ISI Kef est l’une des établissements universitaires de l’université de

Jendouba.

Dénomination : Institut Supérieur d’informatique du Kef (ISI).

Directeur : Mr Hayouni Mohamed.

Secteur : général : Mr Khammassi Aissa.

Téléphone : 78201056

Fax : 78200237

Site web : [www.isikef.rnu.tn](http://www.isikef.rnu.tn)

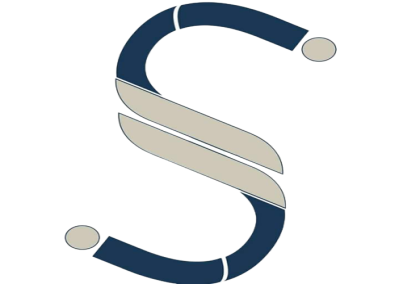


Figure 1 : Logo <ISI KEF >

**1.2. Liste de diplômes**

Les formations assurées par L’ISI Kef sont :

* Licence en Sciences de l’Informatique.
* Licence Appliquée en Technologie du Multimédia et du Web.
* Licence Appliquée en Technologies des Réseaux Informatique.
* Mastère de Recherche en Système d’Informatique et Web.
* Mastère Professionnel en Administration et Sécurité des Réseaux Informatique.
* Mastère Co-construite en nouvelles Technologies de l’Information et de la Communication dédiées à l’Innovation de l’Agriculture.
* Mastère Professionnel en Multimédia et Web.

**1.3. Organigramme de la société**

Un organigramme est la représentation graphique de la structure d’un organisme, d’un établissement ou d’un service avec ses dévers élément et leurs relations.

Directeur

**Secrétaire**

**Principal**

**Conseil**

**Scientifique**

**Scolarité**

**Bibliothèque**

**Service des**

**Examens**

**Service**

**Informatique**

**Service**

**Administratif**

**Administratif**

**Service**

**Financier**

Figure 2 : Organigramme de l’organisme d’accueil

**2. présentation du projet**

**2.1. Description du projet**

Ce projet rentre dans le cadre du projet de fin d’études qui vient conclure notre formation à l’Institut Supérieur d’Informatique du Kef. Nous avons réalisé notre stage de projet fin d’études au sien de l’entreprise « ISI Kef ». Notre projet a été réalisé du 05 Février au 15 Mai 2021. La plateforme qu’il a demandée pour la gestion des activités administratives pour l’institut. Elle a pour objectif la conception et le développement d’une plateforme permettre d’offrir des fonctionnalités indispensables à la connaitre les services offrir pour l’étudiant.

L’origine de ce sujet était une simple idée pour fournir des informations concises et pertinentes sur les activités facilement accessibles en cas de besoin. Au fur et à mesure cette idée a évolué pour concevoir un système de gestion des activités administratif.

L'objectif fondamental du site est de :

* Permettre aux étudiants de :

- Consulter leurs fiches des notes et les emplois du temps

- Consulter les évènements

-Gérer la bibliothèque (télécharger, consulter)

* Mettre à disposition d’admin et l’agent administratif la possibilité de

-Gérer les fiches notes et les emplois du temps,

-Gérer les évènements,

-Gérer la bibliothèque

-Gérer les documents administratifs

Notre projet est géré par un Admin, qui gère les fiches notes et les emplois du temps, les évènements, la bibliothèque et gérer les documents administratifs pour valider ou bloquer un étudiant et consulter les statistiques.

Ce type d’application métier s’avère très utile non seulement afin de subvenir aux besoins des utilisateurs mais il peut aussi représenter un réel avantage pour les services. Il s’agit de rapprocher le service de ses utilisateurs et de le rendre plus disponible et surtout accessible en cas de besoin.

La mise en place de **plateforme** nécessite le développement de trois axes à savoir :

* Axe Gestion du projet : Utilisation de la méthode agile SCRUM.
* Axe modélisation conceptuelle : Utilisation du langage de modélisation UML (Unified Modeling Language).
* Axe de développement : l’utilisation de la plateforme de développement Flask.

**2.2. Etude de l’existant**

L’étude de l’existant est une phase importante pour bien comprendre le système de travail actuel, C’est une étape fondamentale pour le développement des applications et permet de situer le projet dans son contexte.

Cette étude se base essentiellement sur les recherches et les études réalisées lors de notre stage chez ISI Kef.

Dans le cadre de connaitre les services fournis par l’ISI, on constate :

* Manque d ‘information
* Document non disponible a distant
* L’étudiant donne sa carte d’identité pour prendre un rapport a une durée limitée
* L’étudiant doit être présent à l’institut pour consulter leur note et son emploi de temps

Avec cette gestion manuelle de fournit de service, il y a le risque de perdre le temp puisque les demandes vont prendre beaucoup de temps pour être validé.

Pour avoir une vision évidente du service et s'assurer d'un bon résultat de l'application, c'est remarquable de faire quelques recherches sur les solutions existantes qui partagent le même principe que notre application « service administratif» pour mettre en valeur leurs qualités et leurs défauts.

**2.3. Critique de l’existant**

Parmi les inconvénients des solutions existantes :

* Difficulté de la gestion manuelle des services administratifs :

-L’utilisation des papiers pour gérer les services complique encore plus la tâche de suivi des demandes.

* Perte de temps

-Le fait de gérer manuellement les prises les services administratifs risque de perdre beaucoup de temps au niveau de la traçabilité et le suivi des demandes.

* Taux d’erreurs de la gestion des prises services administratifs est élevé.
* Perte de la demande d’étudiant :

-En raison du travail toutes les journées, beaucoup des étudiants ne trouvent pas le temps de se déplacer sur place pour prendre des copies des rapports donne sa carte d’identité a une durée limitée, en plus pendant cette période à cause de la propagation de virus Corona, les étudiant évitent beaucoup de quitter leur domicile et être présent à l’institut pour consulter leur note et son emploi de temps.

**2.4. Solution proposée**

On propose une solution dynamique qui assure la gestion des étudiants de l’ISI et les permets de connaitre tous les services fournis de cette société avec la possibilité de prendre ces services en ligne.

Le site web développé se présentera comme une plateforme ou les professionnels de tous types d’activités pourront proposer leurs services et un planning de prise ces services en ligne.

Notre solution consiste à développer une plateforme bien organisée pour gérer les comptes des étudiants. Cette application va traiter la gestion des événements, la bibliothèque, des fiches note w les emplois de temps et les documents administratifs.

L’administrateur a un accès pour gérer utilisateur et gérer l’accès à l’agent administratif il peut accéder facilement à la liste des utilisateurs puis l’accepter ou le bloquer et consulter les statistiques.

L’agent administratif a un accès pour gérer événements, la bibliothèque, des fiches note w les emplois de temps et les documents administratifs.

L’étudiant à un accès pour accéder à la plateforme, entrer ces coordonnés, faire l’inscription, gérer leur propre demande, consulter les événements, les fiches note et les emplois de temps aussi a la possibilité de gérer la bibliothèque de télécharger un rapport ou le consulter très facilement avec la possibilité de le mettre à jour quotidiennement.

**3. Méthode de développement**

La finalisation du projet dans les délais de livraison est le souci majeur de chaque équipe de développement d’un logiciel. L’un des problèmes les plus fréquemment affrontés lors de la construction du logiciel est la mauvaise spécification et le changement brusque des besoins. Cela peut influencer non seulement l’équipe de développement en créant un environnement de stress, mais aussi le temps consacré pour la réalisation du projet et donc des délais de livraison dépassées afin d’éviter ces situations critiques, nous adoptons la méthodologie agile pour la gestion de notre projet.

**3.1. Méthode Agile**

Les problématiques précédemment mentionnées ont poussé les informaticiens à réinventer les méthodes de gestion de projet et de conception en introduisant ce qu’on appelle les méthodes agiles. C’est une approche incrémentale et itérative, menée dans un esprit collaboratif, avec juste ce qu’il faut de formalisme. Elle peut générer un produit de bonne qualité tout en prenant en compte l’évolution des besoins des étudiants. En suivant cette approche, le logiciel est conçu dans un ensemble et peut être construit étape par étape.

**3.2. Choix de la méthodologie de développement**

La méthodologie est un procédé adopté afin de nous formaliser les étapes à suivre pour le développement du projet et pour que ce dernier répond d'une manière efficace aux demandes du client. Dans cette optique, nous avons choisi d'utiliser la méthodologie agile "Scrum" au cours de notre projet.

Notre choix n'est pas au hasard car cette méthodologie choisie présente des avantages spécifiques par rapport aux méthodes classiques. Dans ce contexte, nous citons donne les principaux avantages trouvés qui répondent aux nos objectifs souhaités :

La flexibilité de cette méthode présente son principal avantage surtout la souplesse avec laquelle elle supporte les changements apportés au projet,

Le second avantage de Scrum est l'amélioration de la productivité, Simplicité des processus, Elle permet d'adapter le logiciel crée suivant l'évolution du projet.

**3.3. Présentation des Rôles de « SCRUM »**

Scrum est considéré comme un cadre ou « Framework » de gestion de projet. Ce cadre est constitué d'une définition des rôles, il s'articule autour des trois rôles qui sont principalement les suivants :

* **Product Owner** : Il représente à la fois le client et les utilisateurs. C'est donne lui qui définit les attentes et les besoins du projet. Ainsi, il définit les tâches permettant de répondre à ces besoins et il mettra en place leur priorisation.
* **Scrum Master** : Le Scrum Master assure globalement le bon déroulement des programmes et protège l'équipe de tout problème extérieur. Il assure également l'organisation des réunions et la bonne application de la méthode agile.
* **Equipe ou Team Members :** Ce sont les personnes chargées de la réalisation du Sprint. Elle est composée des professionnels et cares anisée par une forte coopération et une haute communication entre les différents membres.

**4. Processus Scrum**

Le cycle de vie de la méthode SCRUM se décompose en plusieurs Sprints successifs, la figure suivante présente le processus de développement d'un projet selon SCRUM. [1]

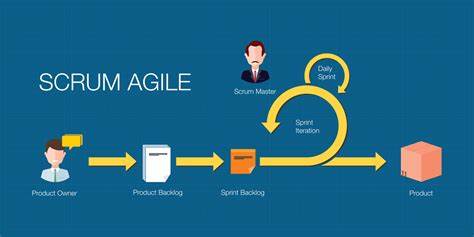


Figure 3 : Processus Scrum

**Voici quelques points détaillant ce schéma :**

L'équipe de développement est pilotée par le Scrum Master.

Scrum est centré sur des itérations, des périodes courtes de développement dont les objectifs dont définis à l'avance, appelées sprints.

Les objectifs font partis d'un référentiel d'exigences appelé le ProductBacklog qui est fourni et tenu à jour par le ProductOwner.

Avant chaque sprint, les fonctionnalités les plus prioritaires passent dans le sprint Backlog et deviennent donc les objectifs à réaliser durant le sprint.

Un sprint démarre toujours par sa planification en partant de discussions entre le ProductOwner et l'équipe concernant le ProductBacklog.

• Chaque sprint améliore la valeur ajoutée du produit en ajoutant de nouvelles fonctionnalités qui peuvent être livrées au client.

**4.1. Approche Agile et Séquentielle**

Le choix de méthodologie se devise en deux. D’abord il s’agit de l’adoption d’une approche classique ou agile. Ensuite, pour bien trancher, il nous faut étude comparative mettant en relief les différences entre les deux approches agiles

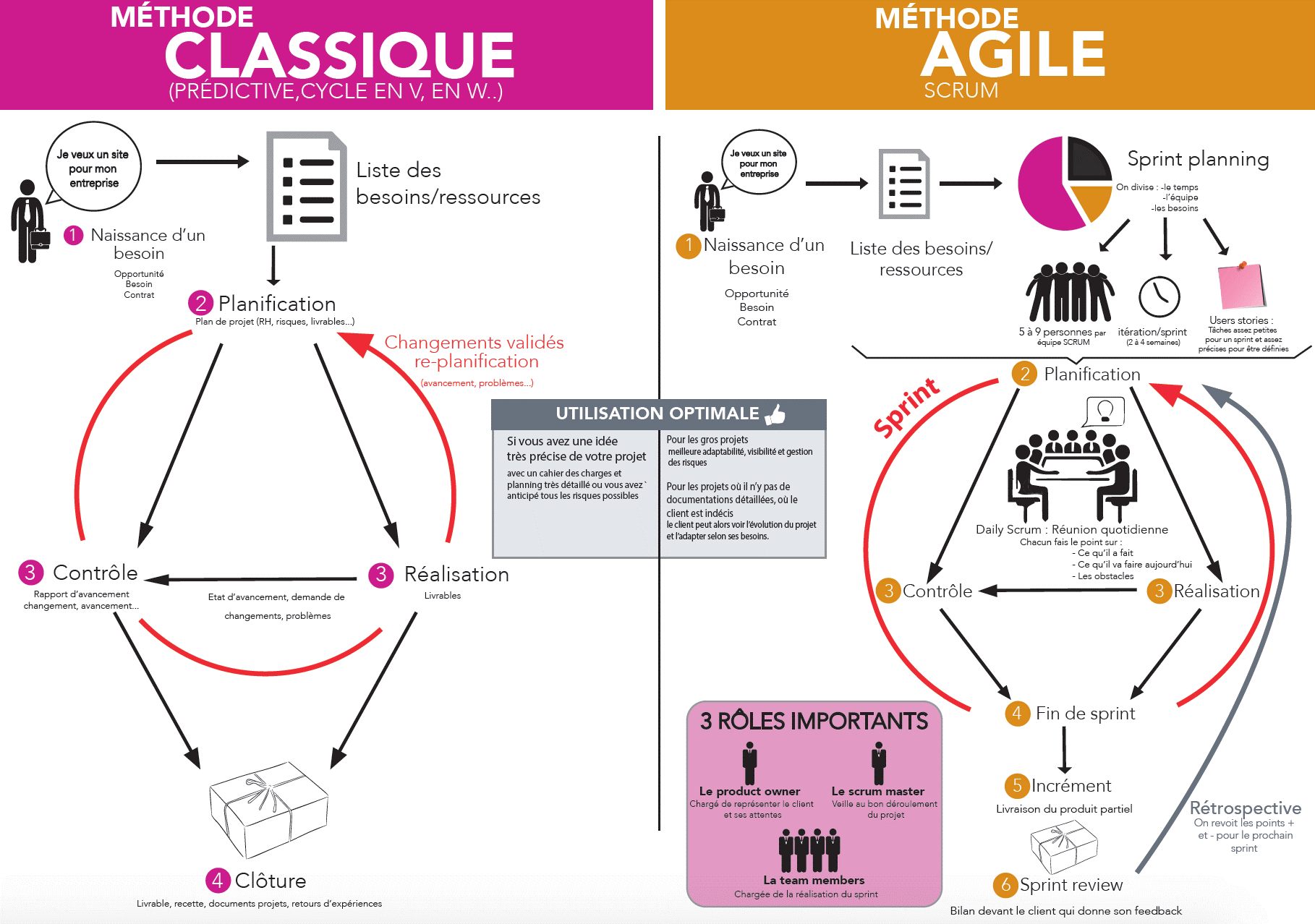


Figure 3 : Comparaison entre les méthodes agile et classique [1]

|  |  |  |
| --- | --- | --- |
| **Thème** | **Approche Traditionnelle** | **Approche Agile** |
| Cycle de vie | En Cascade ou en V, sans rétroaction possible, phases séquentielles. | Itératif et incrémentale |
| Planification | Prédictive, caractérisés par les plans au moins détaillés d’un périmètre et des exigences définies et stables au début du projet. | Adaptive avec plusieurs niveaux de planification (macro et micro planification) avec agissements si nécessaires au fil de l’eau en fonction des changements survenus. |
| Documentation | Produite en quantité importante comme support de communication, de validation et de contractualisation. | Réduite au strict nécessaire au profit d’incréments fonctionnels opérationnels pour obtenir le feedback du client. |
| Equipe | Une équipe avec des ressources spécialisées, dirigée par un chef de projet. | Une équipe responsabilisée ou l’initiative et la communication sont privilégiées, soutenu par le chef de projet. |
| Qualité | Contrôle de qualité à la fin du cycle de développement. Le client découvre le produit fini. | Un contrôle de la qualité précise et permanente, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment. |
| Changement | Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés. | Accueil favorable au changement inéluctable intégré dans le processus. |
| Suivi de l’avancement | Mesure de la conformité aux plans i initiaux. Analyse des écarts. | Un seul indicateur  d’avancement: le nombre des fonctionnalités implémentés et le travail restant à faire. |
| Gestion des risques | Processus distinct, rigoureux de gestion des risques. | Gestion des risques intégrée dans le processus global avec responsabilisation des chacun dans l’identification et la résolution des risques. Pilotages des risques. |
| Mesure de succès | Respect des engagements initiaux aux termes de cout, de budget et de niveau de qualité. | Satisfaction client par la livraison de valeur ajouté. |

Tableau : Différence entre approche traditionnelle et approche agile

**2.4. Langage de modélisation adopté**

En regardent les objectifs fixés pour la réalisation du projet, nous remarquons que nous sommes face à une application modulaire et qui devra rester ouverte pour les futures améliorations. De ce fait, il est important d’utiliser un langage universel pour la modélisation.

Notre choix est porté sur le langage UML (Unified Modeling Langage) qui a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l’architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.



Figure 4: Logo UML

**5. Conclusion**

Tout au long de ce chapitre, nous avons présenté l'organisme **ISI Kef** et ses principales activités. Par ailleurs, nous avons pu dégager le contexte général du projet et présenter le choix de la méthodologie de développement. En définissent le champ de notre étude suivi d’une étude de l’existant afin de préciser les objectifs à atteindre. En effet, l’étude de l’existant nous a permis de préparer une bonne conception pour les améliorations que nous allons ajouter dans la solution proposée afin de répondre à nos besoins.

# Chapitre 2 : Etude préalable

## Introduction :

## Avant de démarrer un projet, il est nécessaire de le définir et de le planifier afin de bien le piloter et d’atteindre l’objectif de client. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités qui doivent répondre aux attentes des différents utilisateurs de notre application, ils doivent être persistants spécifiques et réalisables et les besoins fonctionnels qui représentent les exigences implicites auxquels le système répondre.

## Spécification des besoins :

### 1. Les Besoins fonctionnels

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande.

* **Authentification :** Chaque utilisateur (étudiant, agent administratif, admin), possède un login et un mot de passe spécifique qui lui permet de vérifier son identité.
* **gérer utilisateur :** l’application doit donner la possibilité à l’acteur principal admin d’ajouter, modifier, supprimer, ou afficher un utilisateur.
* **gérer les fiches note et les emplois de temps :** l’application doit donner la possibilité à l’acteur agent administratif d’ajouter, modifier, supprimer, ou afficher les fiches des notes et les emplois de temps
* **gérer les événements :** l’application doit donner la possibilité à l’acteur agent administratif d’ajouter, modifier, supprimer, ou afficher les événements dans la société
* **gérer concours master :** l’application doit donner la possibilité à l’acteur admin d’ajouter, modifier, supprimer, ou afficher les concours master
* **gérer les statistiques :** l’application doit donner la possibilité à l’acteur principal l’admin de gérer les statistiques.
* **gérer la bibliothèque :** l’application doit donner la possibilité à l’acteur agent administratif d’ajouter, modifier, supprimer, ou afficher les rapports pfe, pfa et les livres dans la bibliothèque et permet aux étudiants troisième année d’ajouter une copie numérique de leur rapport.
* **gérer les documents administratifs :** l’application doit donner la possibilité à l’acteur agent administratif d’ajouter, modifier, supprimer, ou afficher les documents administratifs
* **gérer les classe :** l’application doit donner la possibilité à l’acteur Admin d’ajouter, modifier, supprimer, ou afficher les classes
* **gérer les matières :** l’application doit donner la possibilité à l’acteur Admin d’ajouter, modifier, supprimer, ou afficher les matières
* **S’inscrire :** l’application doit donner la possibilité à l’internaute d’accéder au notre application et faire l’inscription
* **Consulter les événements :** l’application doit donner la possibilité à l’admin et l’étudiant de consulter les événements dans l’ISI
* **Consulter les fiches des notes et les emplois de temps :** l’application doit donner la possibilité à l’admin et l’étudiant de consulter lesfiches des notes et les emplois de temps

**2. Les besoins non fonctionnels :**

Ce sont des exigences qui ne concernant pas spécifiquement le comportement du système mais plutôt identifiant des contraintes internes et externes du système.

* **La sécurité** : la sécurité informatique est l’ensemble des moyens techniques pour garantir la sécurité des logiciels informatiques. Parmi ces techniques on cite l’authentification qui nous permet de définir les privilèges de chaque utilisateur.
* **La fiabilité** : le système doit fonctionner de façon cohérence sans erreurs et doit être correcte.
* **La performance** : l’application devra être performante c’est-à-dire que le système doit répondre dans un délai acceptable.

## Identifications des acteurs :

## Un acteur représente un rôle joué par une entité qui interagisse immédiatement avec le système étudié.

* **Admin**: Il est responsable de son profil et peut gérer les utilisateurs, les concours master, les statistiques et consulter les événements, les fiches des notes , les emplois de temps .
* **Agent Administratif** : Il est responsable à gérer les événements, les fiches des notes, les emplois de temps .et la bibliothèque
* **Etudiant** : Son rôle de consulter événements, les fiches des notes, les emplois de temps .et de gérer la bibliothèque (télécharger , consulter ou ajouter ) .

## Etude technique

### 1. Environnement de travail :

Dans cette partie nous étudions notre choix des outils matériels et logiciels du développement de notre applications.

#### Environnement matériel :

J’indique les caractéristiques de mon ordinateur sur lesquelles je développe l’application.

Notre plateforme a été développée par un ordinateur portable ACER



Figure 5: PC ACER Intel(R) Core (TM) i3-7020U

|  |  |
| --- | --- |
| Machine | ACER Series 00425 |
| Propriétaire | ACER |
| Processeur | I3-7020u |
| RAM | 4GB |
| Disque Dur | 1TB |
| Processeur | 64bits |
| Système d’exploitation | Windows 11 pro |

Tableau : ACER série 00425

#### Environnement logiciel :

* **Outil de conception de projet :**

**StarUML :** c’est un logiciel de modélisation UML open source qui peuvent remplacer des logiciels commerciaux et coûteux comme Rational Roisel. Étant simple d’utilisation nécessite peu de ressources système, supportant UML2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation et nous l’avons utilisé pour créer nos diagrammes.

Figure : Logo de StarUML

* **Outil de développement :**

**-Flask** : est un **micro Framework open-source** de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires.

****

Figure : Logo de Flask

**-PyCharm :** est un**environnement de développement intégré utilisé pour programmer en Python**. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions.

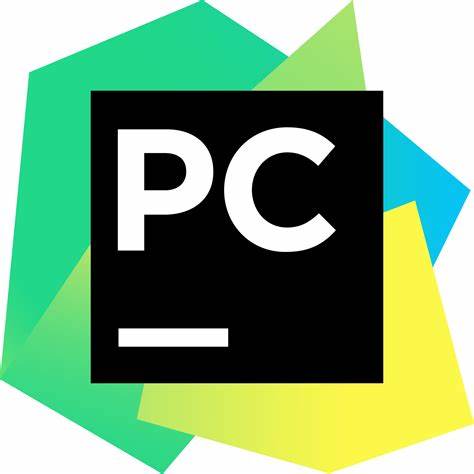


Figure : Logo de PyCharm

### 2. Langage de programmation

**-JavaScript :** est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs comme Node.js. C’est un langage orienté objet à prototype et notamment une propriété de prototypage



Figure : Logo de JavaScript

**-Python** : est un **langage de programmation interprété, multi- paradigme et multiplateformes, il** favorise la programmation impérative structurée, fonctionnelle et orientée objet.



Figure : Logo de Python

**-Bootstrap**: est une **infrastructure de développement frontale**, open source pour la création de sites et d'applications Web.



Figure : Logo de Bootstrap

**HTML et CSS** : sont deux langues principales couramment utilisées pour le développement Web. La différence entre HTML et CSS réside dans le fait que HTML est un langage de balisage utilisé pour créer la structure d'une page Web, tandis que le CSS est un langage de style utilisé pour rendre les pages Web plus présentables.



Figure : Logo de HTML et CSS

### 3. Serveur d’application :

**Xampp :** est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s’agit d’une distribution, de logiciels libres offrant une bonne souplesse d’utilisation, réputée pour son installation simple et rapide



Figure : Logo de Xampp

### 4. Architecture du système :

Le pattern modèle-vue-contrôleur (en abrégé MVC, de l’anglais Model-View-Controller), est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Ses avantages :

* Séparation des compétences (design, base de données, application)
* Simplicité de mise à jour
* Vitesse de création de pages.

Ce paradigme regroupe les fonctions nécessaires en trois catégories :

* Un modèle (Modèle de données) ;
* Une vue (présentation, interface utilisateur) ;
* Un contrôleur (logique de contrôle).

Nous expliquons ces trois parties l’une après l’autre :

* **Model (**Modèle en français**) :** Le modèle représente les structures de données. Typiquement, les classes modèles contiennent des fonctions qui aident à récupérer, à insérer et à mettre à jour des informations de la base de données.
* **View (**vue en français**) :** La vue correspond à l'interface avec laquelle l'utilisateur interagit. Elle se présente sous la forme d'une Template représentant l'interface

Reprenons l'exemple de l’outil. Ce n'est pas le contrôleur qui affiche le formulaire, il ne fait qu'appeler la bonne vue.

* **Controller (**Contrôleur en français**) :** Il gère l'interface entre le modèle et l’utilisateur. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

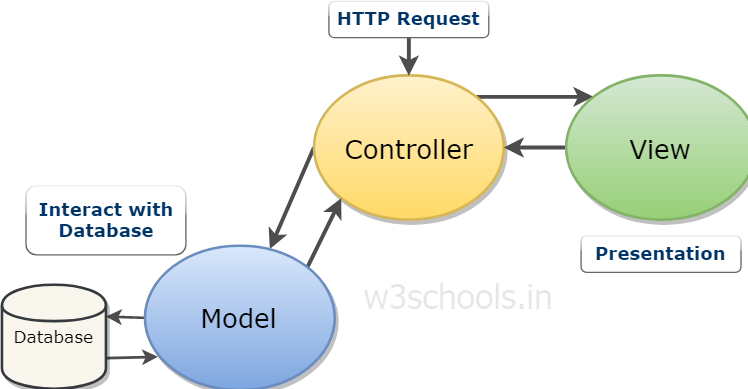


Figure 17: Le patron MVC

## Préparation et mise en œuvre de sprint 0 :

### Les acteurs du sprint :

Le principe de base de Scrum est de se focaliser d’une façon itérative sur l’ensemble de fonctionnalités à réaliser dans chaque sprint. Et comme nous avons définis dans le chapitre 1 que SCRUM se base sur 3 rôles principaux : Le Product Owner, Le SCRUM Master et le SCRUM Team.

Dans notre projet les rôles sont répartis comme suit :

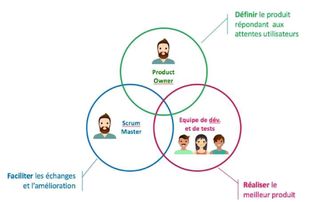


Figure : Hiérarchie de l'équipe

### Les « user stories » :

Dans les méthodes agiles, un « user story » est une phrase simple dans le langage de tous les jours permettent de décrire avec suffisamment de précision le contenu d’une fonctionnalité à développer. La phrase contient généralement trois éléments descriptifs de la fonctionnalité :

Qui ? Quoi ? Pourquoi ? En tant que « qui », Je veux « quoi » afin de « pourquoi »

Chaque « user story » sera estimée et priorisée mais ce qui caractériser aussi une « user story», c’est qu’elle doit être suffisamment petite être livrée dans un sprint

**Priorités**

Le choix de priorités d’une étape se base sur l’influence et l’importance d’une tache sur le système. On peut diviser les priorités en sous priorité haute, moyenne et faible. Le choix de priorités dans cette section s’est basé sur la dépendance entre les fonctionnalités de l’application.

**Risque**

Lors du pilotage d’un projet, l’identification des risques critique présente une étape indispensable pour la réussite de ce dernier. Pour notre cas, le seul risque qui peut nous ralentir est lié à la complexité de l’application et à différentes contraintes à respecter

### 3. Backlog produit :

Le Backlog de produit est un artéfact très important dans Scrum, c’est l’ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Les caractéristiques fonctionnelles sont appelées des histoires utilisateur (User Story) et les caractéristiques techniques sont appelées des histoires techniques (Technical Story) Le tableau ci-dessous résume le Backlog produit de notre application :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | | **Thème** | **Nom** | **Acteur** | **Description (User Stories)** | **Risque** | **Priorité** |
| 1 | Authentification | | S’authentifier en tant qu’Admin | Admin | Entant qu’admin je dois m’authentifier afin d’accéder à ma session | Faible | Elevée |
| S’authentifier en tant que Agent | Agent-  Administratif | Entant que qu’Agent dois m’authentifier afin d’accéder à ma session. | Faible | Elevée |
| S’authentifier en tant que Etudiant | Etudiant | Entant que Etudiant je dois m’authentifier afin d’accéder à ma session. | Faible | Elevée |
| 2 | Gérer les utilisateurs | | Consulter la liste des utilisateurs | Admin | Entant qu’admin je veux visualiser leurs informations | Faible | Elevée |
| Ajouter utilisateur | Admin | Entant qu’admin je veux ajouter un utilisateur | Faible | Elevée |
| Modifier utilisateur | Admin | Entant qu’admin je veux changer le statut d’un fournisseur. | Faible | Elevée |
| Supprimer utilisateur | Admin | Entant qu’admin je veux supprimer utilisateur | Faible | Elevée |
| Détails d’utilisateur | Admin | Entant qu’admin je veux consulter plus de détails sur les utilisateurs | Faible | Elevée |
| Chercher un utilisateur | Admin | Entant qu’admin je veux trouver un utilisateur spécifique. | Faible | Elevée |
| 3 | Consulter les statistiques | | Consulter la page d’accueil | Admin | Entant qu’admin je veux visualiser les statistiques relatives à l’application. | Faible | Elevée |
| 4 | Consulter les événements, la bibliothèque, les documents administratif | | Consulter la liste des événements | Admin | Entant qu’admin je veux visualiser les listes d’événement, la bibliothèque, les documents administratif relatives à l’application. | Faible | Elevée |
| 5 | Gérer le profil | | Modifie mot de passe | Admin | Entant qu’admin je veux modifier mot de passe | Faible | Elevée |
| 6 Gérer les classes | | | Ajouter une classe | Admin | Entant qu’admin je veux ajouter une classe | Faible | Elevée |
| Modifier une classe | Admin | Entant qu’admin je veux modifier une classe | Faible | Elevée |
| Supprimer une classe | Admin | Entant qu’admin je veux supprimer une classe | Faible | Elevée |
| 7 Gérer les matières | | | Ajouter une matière | Admin | Entant qu’admin je veux ajouter une matière | Faible | Elevée |
| modifier une matière | Admin | Entant qu’admin je veux modifier une matière | Faible | Elevée |
| supprimer une matière | Admin | Entant qu’admin je veux supprimer une matière | Faible | Elevée |
| 8 | Gestion des événements | | Créer un nouvel événement | Agent administratif | Entant qu’Agent administratif je veux créer un nouvel événement | Faible | Elevée |
| Modifier un événement | Agent administratif | Entant qu’Agent je veux mettre à jour les informations d’un événement | Faible | Elevée |
| Supprimer événement | Agent administratif | Entant qu’fournisseur je veux supprimer des événements | Faible | Elevée |
| 9 | Gérer la bibliothèque | | Ajouter un rapport, livre … | Agent administratif | Entant qu’fournisseur je veux accepter des rapports. | Faible | Elevée |
| modifier un rapport, livre … | Agent administratif | Entant qu’agent je veux changer le statut d’un rapport | Faible | Elevée |
| 10 |  | | afficher un rapport, livre | Agent administratif | Entant qu’fournisseur je veux régler afficher un rapport livre … | Faible | Elevée |
| 11 | Gérer les documents administratifs | | Ajouter, modifier, supprimer, afficher | Agent administratif | Entant qu’agent je veux Ajouter, modifier, supprimer, afficher les documents | Faible | Elevée |
| 12 | Gérer le profil | | Modifier mot de passe | Agent  Administratif | Entant qu’agent je veux mot de passe | Faible | Elevée |
| Modifier coordonnées | Agent administratif | Entant qu’agent je veux modifier coordonnée | Faible | Elevée |
| 13 | Consulter les événements | | Recherche par Nom | Etudiant | Entant que client je veux rechercher un fournisseur par un nom | Faible | Elevée |
| Recherche par spécialité | Etudiant | Entant que client je veux rechercher un fournisseur par spécialité | Faible | Elevée |
| Recherche par adresse | Etudiant | Entant que client je veux rechercher un fournisseur par adresse | Faible | Elevée |
| Prendre rendez-vous | Etudiant | Entant que client je veux prendre un rendez-vous | Faible | Elevée |
| 14 | Consulter la bibliothèque,  Les notes, les emplois de temps , les événements | | Consulter la liste des rapports, livres … | Etudiant | Entant qu’Etudiant je veux Visualiser la bibliothèque | Faible | Elevée |
| Consulter les notes | Etudiant | Entant que qu’Etudiant je veux Visualiser les notes | Faible | Elevée |
| Consulter les événements | Etudiant | Entant que qu’Etudiant je veux Visualiser les événements | Faible | Elevée |

Tableau : Le Backlog Product

### Diagramme de cas d’utilisation globale :

Le diagramme de cas d’utilisation général offre une vue globale des différents services offerts par une application. Il expose une vision générale des fonctionnalités assurées par notre plateforme vis-à-vis de l’utilisateur. Pour avoir un diagramme clair et lisible, nous avons subdivisé ce diagramme conformément aux fonctionnalités de ces acteurs que nous détaillons dabs ce qui suit

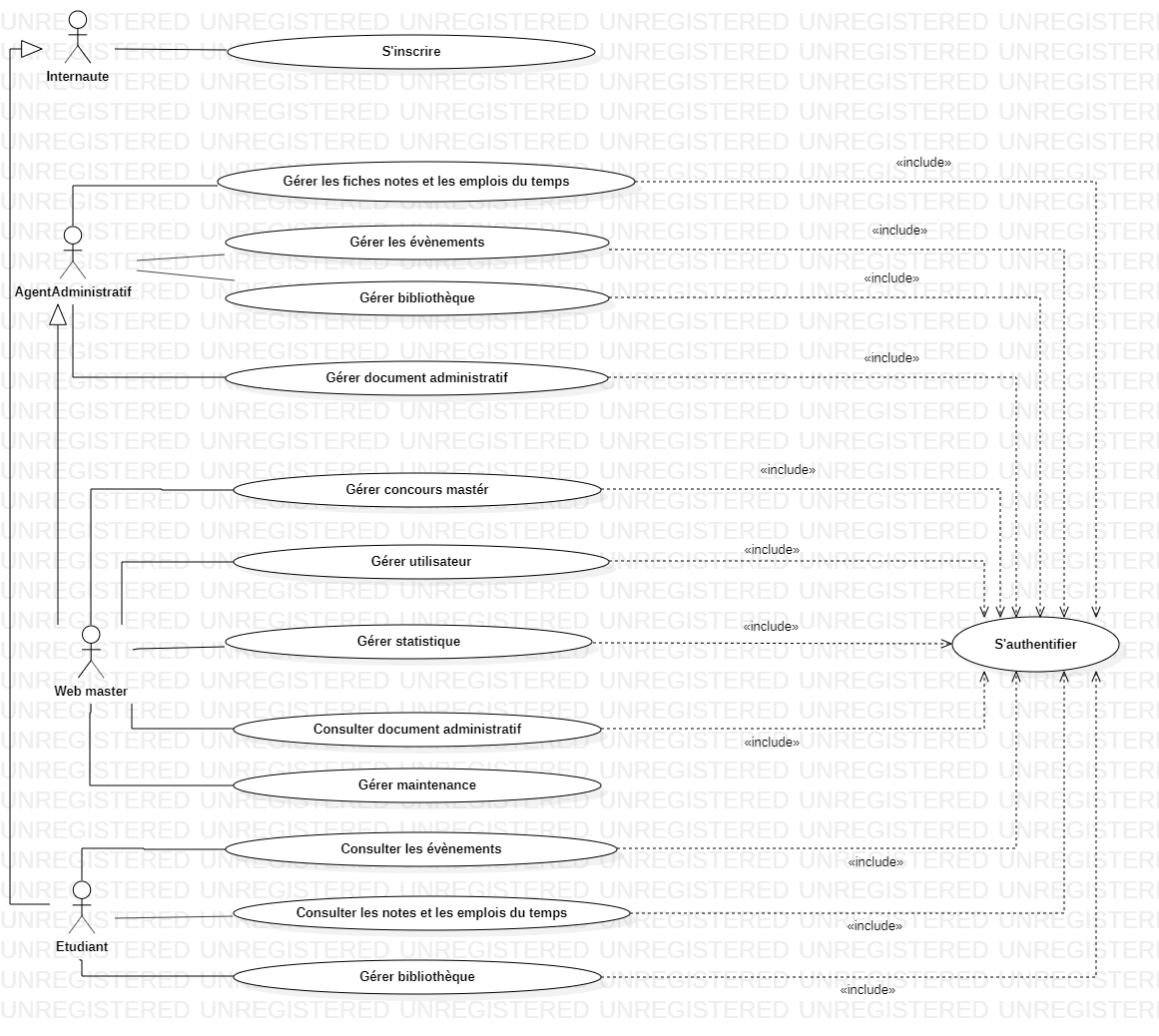


Figure 0: Diagramme de cas d'utilisation globale

**5. Planification des sprints :**

Le travail sera planifié selon des sprints que nous avons définis et chacun dure à 3 semaines. Après une réunion avec le Scrum master, on a identifié 3 sprints. Nous avons utilisé un diagramme qui nous permettrons de bien planifier les sprints nommée « Gantt ».

………………………………………………………….

Figure : Planification des sprints

La planification des sprints est ainsi schématisée dans le cycle de vie suivant :

…………………………………………………

Figure : Backlog produit

## Conclusion :

Tout au long de ce chapitre, nous avons ensuite présenté les besoins fonctionnels et non fonctionnels pour avoir une vision plus claire du sujet, Dans la dernière partie notre but était de présenter l'environnement du travail dans un premier lieu, nous avons tour à tour présenté les outils de notre environnement matériels et logiciels, puis nous avons cité les langages de programmation et la technologie ainsi l'architecture que nous avons utilisée. Sans oublier que ce chapitre nous a permis de planifier et organiser le temps consacré à la réalisation du projet en identifiant les sprints, les acteurs et le Backlog produit et la description du langage objet unifié. Dans le chapitre suivant nous allons détailler le sprint du projet.

# **Chapitre3 : Implémentation des fonctionnalités de l'acteur Admin**

## Introduction

Le terme release peut-être défini comme une période de temps qui permet de produire une version distribuée d’une application. Un release est contribué d’une suite d’itération (sprint) qui se termine quand les incréments de ces derniers construisent un produit.

Dans ce chapitre, nous allons nous intéresser à la partie web système, de l’analyse des besoins jusqu’au test de ce derniers, ainsi nous traiterons en détails chacun des cas d’utilisation préalablement présenté

## Premier Sprint :

Dans le chapitre précèdent, nous avons exposé la phase d’initialisation de notre projet. L’étape la plus crucial dans le processus Scrum est le sprint. Celui-ci va être réalisé pendant une certaine tranche de temps et englobera les quatre étapes du cycle de vie Scrum, précisément la spécification fonctionnelle, la conception, le développement et les tests. Notre premier sprint évoque le fait de s’authentifier qui est notre première fonctionnalité dans le Backlog du produit et dont nous dressé sa liste des user stories

Suite à une étude sur la division du projet, nous avons décidé d’atteindre les buts suivants :

Le sprint Backlog est le tableau que nous tirons de Backlog Product qui formalise le calendrier pour le sprint :

### Backlog du Sprint 1 :

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Thème** | **Description** | **Priorité** |
| **1** | S’authentifier | En tant qu’admin je souhaite authentifier pour accéder à mon espace personnel d’application. | Elevé |
| **2** | Vérifier les comptes utilisateur | En tant qu’admin je souhaite valider ou bloquer un utilisateur | Elevé |
| **3** | Gérer le profil | En tant qu’admin je souhaite gérer mon profil | Elevé |
| **4** | Gérer les classes | En tant qu’admin je souhaite gérer les classes | Elevé |
| **5** | Gérer les matières | En tant qu’admin je souhaite gérer les matières | Elevé |
| **6** | Gérer la maintenance | En tant qu’admin je souhaite gérer les maintenances | Elevé |
| **7** | Gérer concours master | En tant qu’admin je souhaite gérer les concours master | Elevé |
| **8** | Consulter les statistiques | En tant qu’admin je souhaite consulter les statistiques | Elevé |
| **9** | Consulter les événements | En tant qu’admin je souhaite consulter la liste événements | Elevé |
| **10** | Consulter les notes et les emplois te temps | En tant qu’admin je souhaite consulter la liste des notes et les emplois te temps | Elevé |
| **11** | Consulter document administratif | En tant qu’admin je souhaite gérer les documents administratifs | Elevé |

1. Tableau : Backlog du sprint 1

### Spécification fonctionnelle :

Dans notre cas la spécification se fait par la présentation du diagramme de cas d’utilisations et la description textuelle de ces derniers.

#### Raffinement du cas d’utilisation « s’authentifier »

L’utilisateur (un admin, un agent administratif, un étudiant) va pouvoir se connecter à notre application en composant son login et son mot de passe, le système doit assurer di son existence avant de lui donner l’accès à l’application.

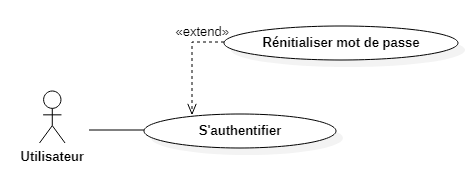


Figure :Diagramme de cas d’utilisation “S’authentifier”

Description textuelle du cas d’utilisation « S’authentifier»

|  |  |
| --- | --- |
| **S’authentifier** | |
| **Cas d’utilisation** | **S’authentifier** |
| Résumé | L’utilisateur doit saisie sont login et mot de passe, le système vérifie les données saisies avant d’afficher l’espace admin. |
| Acteur | Utilisateur |
| **Description des enchainements** | |
| Pré-condition | Post-condition |
| L’utilisateur doit avoir un compte (login et un mot de passe). | Accès à son espace |
| **Scenario principale** | |
| 1. L’utilisateur saisit les paramètres de connexion (email/password) et confirme les paramètres déjà entrés.  2. Le système vérifie les paramètres de connexion (email/password).  3. Le système affiche l’espace de l’utilisateur authentifié. | |
| **Scenario Alternatif(s)** | |
| 1. L’utilisateur saisit mot de passe ou (et) nom d’utilisateur incorrecte(s)  2. Un message d’erreur d’affiche  3. L’enchainement reprend au point 1 du scenario principal.  4. Si L’utilisateur a oublié son mot de passe et veux la réinitialiser, il clique sur le bouton ‘oublié mot de passe’ et le système affiche une interface pour qu’il saisit son email ensuite il envoie un lien dans son email qui permet l’utilisateur de saisir un nouveau mot de passe. | |

Tableau : Description textuelle de cas d'utilisation « s'authentifier »

#### Raffinement du cas d’utilisation « Gérer utilisateur »

L’admin aura le privilège de valider la liste des utilisateurs, il pourra soit ajouter un utilisateur, modifier, supprimer ou consulter de ce dernier.

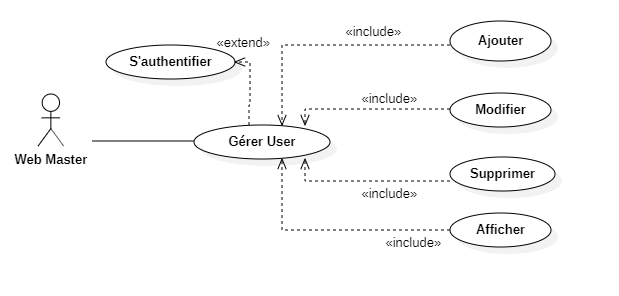


Figure : Gérer utilisateur

Description du cas d’utilisation « Ajouter un utilisateur »

|  |  |
| --- | --- |
| **Gérer utilisateur** | |
| Cas d’utilisation | Ajouter un utilisateur |
| Résumé | L’admin peut Ajouter un utilisateur |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | Un compte utilisateur est ajouté |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des utilisateurs ».  2. le système affiche la page.  3. l’admin choisi le bouton « ajouter ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des utilisateurs » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 6: Description textuelle de cas d'utilisation « ajouter utilisateur »

Description du cas d’utilisation « consulter un utilisateur»

|  |  |
| --- | --- |
| **Gérer utilisateur** | |
| Cas d’utilisation | consulter un utilisateur |
| Résumé | L’admin peut consulter un utilisateur |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | La liste des utilisateurs est consultée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des utilisateurs ».  2. le système affiche la page.  3. l’admin choisi le bouton « Bloquer ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des utilisateurs » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 7: Description textuelle de cas d'utilisation «consulter un utilisateur»

#### Raffinement du cas d’utilisation « Gérer profil »

L’admin aura le privilège de modifier son mot de passe.

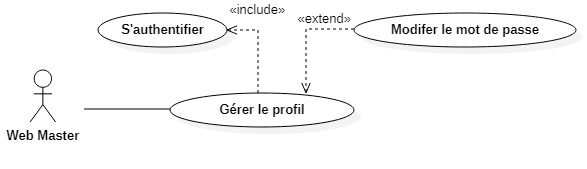


Figure 15: Gérer le profil

Description du cas d’utilisation « Gérer le profil »

|  |  |
| --- | --- |
| **Gérer le profil** | |
| Cas d’utilisation | Modifier mot de passe |
| Résumé | L’admin peut modifier son mot de passe |
| Acteur | Admin |
| **Description ses enchainements** | |
| Pré condition | Post condition |
| L’Admin doit être authentifié | Le mot de passe est modifié |
| **Scénario nominal** | |
| 1. L’Admin demande la page profil  2. le système affiche la page.  3. l’Admin modifier son mot de passe et choisir « modifier mot de passe »  4. le système lui affiche un message de succès. | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. les coordonnées saisies sont incorrectes.  4. Le système indique qu’il y a un champ vide ou incorrect.  L’enchainement démarre au point 3 du scénario nominal | |

Tableau 8: Description textuelle de cas d'utilisation «Gérer le profil»

#### Raffinement du cas d’utilisation « Gérer maintenance »

L’admin aura le privilège Gérer la maintenance.

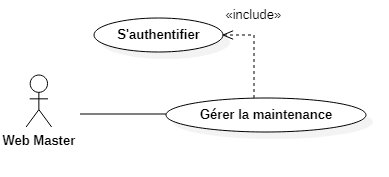


Figure16 : Gérer la maintenance

Description du cas d’utilisation « Gérer la maintenance »

|  |  |
| --- | --- |
| **Gérer la maintenance** | |
| Cas d’utilisation | Gérer la maintenance |
| Résumé | L’admin peut gérer la maintenance |
| Acteur | Admin |
| **Description ses enchainements** | |
| Pré condition | Post condition |
| L’Admin doit être authentifié | Corriger l’erreur |
| **Scénario nominal** | |
| 1. L’Admin demande la page gérer la maintenance  2. le système affiche la page.  3. l’Admin Corrige l’erreur  4. le système lui affiche un message de succès. | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. les coordonnées saisies sont incorrectes.  4. Le système indique qu’il y a un champ vide ou incorrect.  L’enchainement démarre au point 3 du scénario nominal | |

Tableau 9: Description textuelle de cas d'utilisation «Gérer maintenance »

#### Raffinement du cas d’utilisation « Gérer concours master»

L’admin aura le privilège d’ajouter, modifier, supprimer ou afficher un concours master

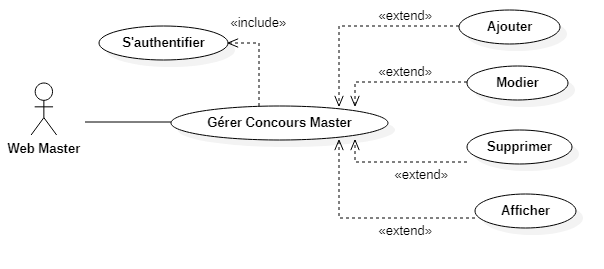


Figure17 : Gérer concours master

Description du cas d’utilisation « Ajouter un concours master »

|  |  |
| --- | --- |
| **Gérer concours master** | |
| Cas d’utilisation | Ajouter un concours |
| Résumé | L’admin peut Ajouter un concours |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | Un concours est ajouté |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des concours».  2. le système affiche la page.  3. l’admin choisi le bouton « ajouter ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des concours» avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 10: Description textuelle de cas d'utilisation «Ajouter concours master»

Description du cas d’utilisation « consulter un concours master >

|  |  |
| --- | --- |
| **Gérer concours master** | |
| Cas d’utilisation | consulter un concours |
| Résumé | L’admin peut consulter un concours |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | La liste des concours est consultée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des concours».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des concours» avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 11: Description textuelle de cas d'utilisation «consulter concours master»

#### Raffinement du cas d’utilisation « consulter statistique »

L’admin aura le privilège d’afficher les statistiques

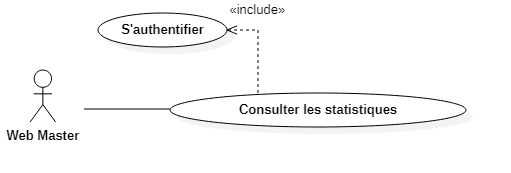


Figure 18 : consulter les statistiques

Description du cas d’utilisation « consulter les statistiques  *»*

|  |  |
| --- | --- |
| **Gérer les statistiques** | |
| Cas d’utilisation | consulter les statistiques |
| Résumé | L’admin peut consulter les statistiques |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | La liste des statistiques est consultée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des statistiques ».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des statistiques » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 12: Description textuelle de cas d'utilisation «consulter les statistiques »

#### Raffinement du cas d’utilisation « consulter les fiches des notes et les emplois du temps »

L’admin aura le privilège d’afficher les fiches des notes et les emplois du temps

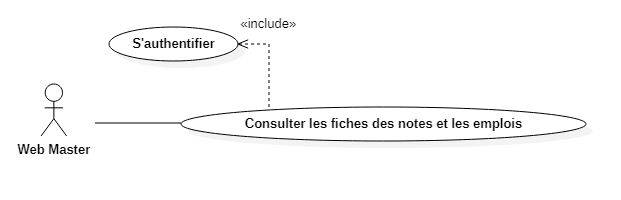


Figure 19 : consulter les fiches des notes et les emplois du temps

Description du cas d’utilisation « consulter les fiches des notes et les emplois du temps *»*

|  |  |
| --- | --- |
| **Consulter les fiches des notes et les emplois du temps** | |
| Cas d’utilisation | Consulter les fiches |
| Résumé | L’admin peut consulter les fiches |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | La liste des fiches est consultée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des fiches ».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des statistiques » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 13: Description textuelle de cas d'utilisation «consulter les fiches »

#### Raffinement du cas d’utilisation « consulter la bibliothèque »

L’admin aura le privilège d’afficher la bibliothèque

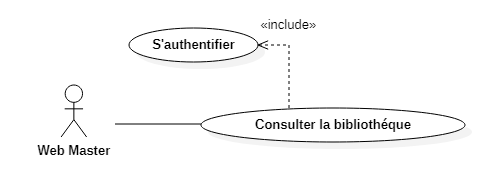


Figure 20: consulter la bibliothèque

Description du cas d’utilisation « consulter la bibliothèque *»*

|  |  |
| --- | --- |
| **Consulter la bibliothèque** | |
| Cas d’utilisation | Consulter la bibliothèque |
| Résumé | L’admin peut consulter la bibliothèque |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | La bibliothèque est affichée |
| **Scénario nominal** | |
| 1. L’admin demande la page « la bibliothèque ».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des statistiques » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 14: Description textuelle de cas d'utilisation «consulter la bibliothèque »

#### Raffinement du cas d’utilisation « consulter les événements »

L’admin aura le privilège d’afficher les événements

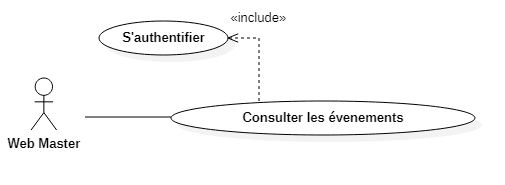


Figure 21: consulter les événements

Description du cas d’utilisation « consulter les événements *»*

|  |  |
| --- | --- |
| **Consulter les événements** | |
| Cas d’utilisation | Consulter les événements |
| Résumé | L’admin peut consulter les événements |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | les événements sont affichés |
| **Scénario nominal** | |
| 1. L’admin demande la page « event ».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste les événements » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 15: Description textuelle de cas d'utilisation «consulter les événements »

#### Raffinement du cas d’utilisation « consulter les documents administratifs »

L’admin aura le privilège d’afficher les documents administratifs

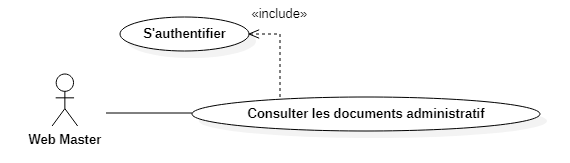


Figure 22 : consulter les documents administratifs

Description du cas d’utilisation « consulter les documents administratifs*»*

|  |  |
| --- | --- |
| **Consulter les documents administratifs** | |
| Cas d’utilisation | Consulter les documents administratifs |
| Résumé | L’admin peut consulter les documents administratifs |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | les événements sont affichés |
| **Scénario nominal** | |
| 1. L’admin demande la page « documents administratifs ».  2. le système affiche la page.  3. l’admin choisi le bouton « afficher ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des documents administratifs» avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler la validation.  Le système retourne au point 2 du scénario nominal. | |

Tableau 16: Description textuelle de cas d'utilisation «consulter les documents administratifs »

#### Raffinement du cas d’utilisation « Gérer classes »

L’admin aura le privilège d’ajouter, modifier, supprimer et afficher les classes

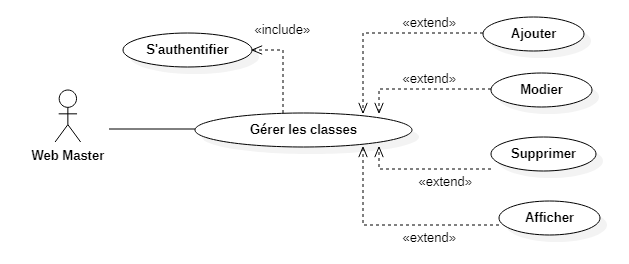


Figure 23 : Gérer les classes

Description du cas d’utilisation « Ajouter une classe »

|  |  |
| --- | --- |
| **Gérer les classes** | |
| Cas d’utilisation | Ajouter une classe |
| Résumé | L’admin peut Ajouter une classe |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | une classe est ajoutée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des classes ».  2. le système affiche la page.  3. l’admin choisi le bouton « ajouter ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des classes » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 17: Description textuelle de cas d'utilisation «ajouter classe »

Description du cas d’utilisation « Modifier une classe »

|  |  |
| --- | --- |
| **Gérer les classes** | |
| Cas d’utilisation | Modifier une classe |
| Résumé | L’admin peut Modifier une classe |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | une classe est modifiée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des classes ».  2. le système affiche la page.  3. l’admin choisi le bouton « modifier ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des classes » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 18 : Description textuelle de cas d'utilisation «modifier classe »

#### Raffinement du cas d’utilisation « Gérer matière »

L’admin aura le privilège d’ajouter, modifier, supprimer ou afficher les matières

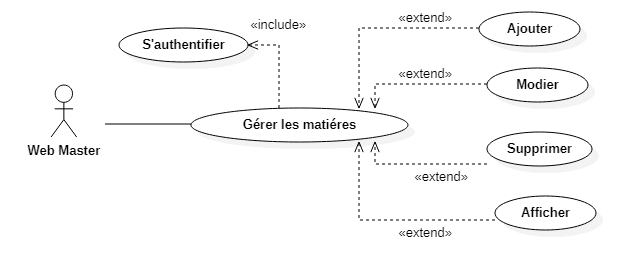


Figure 24 : Gérer matière

Description du cas d’utilisation « Ajouter les matières»

|  |  |
| --- | --- |
| **Gérer les classes** | |
| Cas d’utilisation | Ajouter les matières |
| Résumé | L’admin peut ajouter une matière |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | une matière est modifiée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des matières ».  2. le système affiche la page.  3. l’admin choisi le bouton « ajouter ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des matières » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 19: Description textuelle de cas d'utilisation «Ajouter matière »

Description du cas d’utilisation « Modifier une matière »

|  |  |
| --- | --- |
| **Gérer les classes** | |
| Cas d’utilisation | Modifier une matière |
| Résumé | L’admin peut Modifier une matière |
| Acteur | Admin |
| **Description des enchainements** | |
| Pré condition | Post condition |
| L’admin doit être authentifié | une matière est modifiée |
| **Scénario nominal** | |
| 1. L’admin demande la page « liste des matières».  2. le système affiche la page.  3. l’admin choisi le bouton « modifier ».  4. le système affiche un dialogue de refus.  5. l’admin confirme le refus.  6. le système redirige l’admin vers la page « Liste des matières » avec le nouvel état | |
| **Scénario alternatif(s)** | |
| 1. Problème de connexion au serveur.  2. Problème de connexion à la base de données.  3. l’admin annuler l’ajout.  Le système retourne au point 2 du scénario nominal. | |

Tableau 20: Description textuelle de cas d'utilisation «modifier matière »

1. **Diagramme de séquences détaillées du cas d’utilisation**

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Dans ce qui suit, nous présentons le diagramme de séquence selon les diagrammes de cas d’utilisation de notre système.

#### Diagramme de séquence détaillé du cas d’utilisation « S’authentifier »

Le diagramme exposé ci-dessous explique le scénario d’authentification, l’utilisateur remplit

Le formulaire de saisie de données et confirme l’authentification. Le contrôleur vérifie si les champs sont remplis et les données sont valides puis redirige l’utilisateur vers la page des menus. Dans le cas contraire, le formulaire affiche des messages d’erreur éventuels.

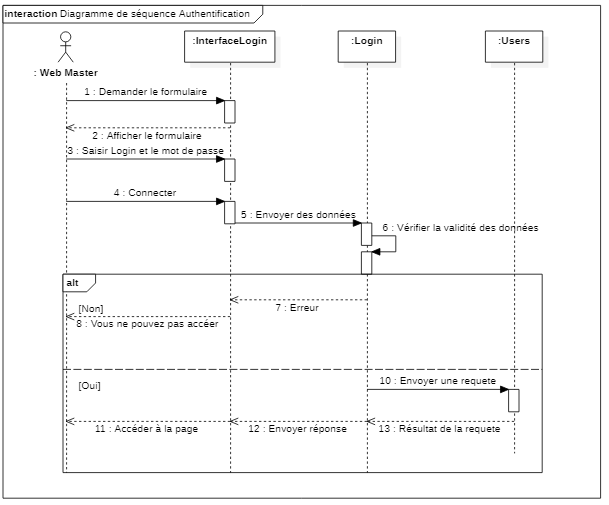


Figure 22: diagramme de séquence détalé de cas d’utilisation authentification

#### Diagramme de séquence détaillé du cas d’utilisation « Gérer utilisateur »

#### Diagramme de séquence du CU« Ajouter utilisateur »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Ajouter un utilisateur ».

Pour ajouter un utilisateur. Le Web Master doit remplir le formulaire d’ajout (nom, prénom, email, téléphone mobile, type, password), puis cliquer sur le bouton « ajouter » afin que le système valide les données, dans le cas où les champs entrés sont valides. Si la saisie est erronée, le système affiche un message d’erreur.

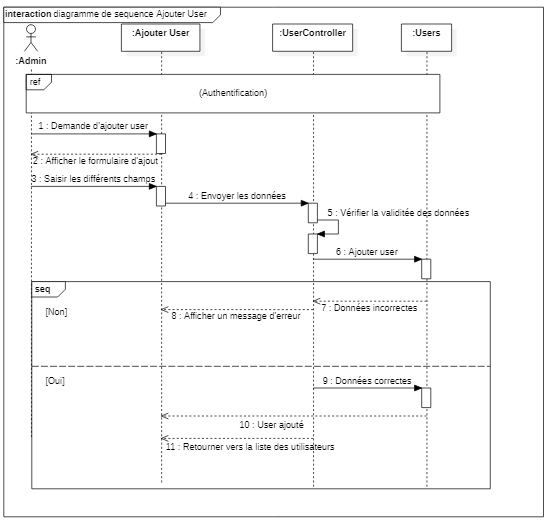


Figure 23: diagramme de séquence détaillé du cas d’utilisation « Ajouter utilisateur »

* **Diagramme de séquence du CU« Modifier utilisateur »**

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Modifier utilisateur ».

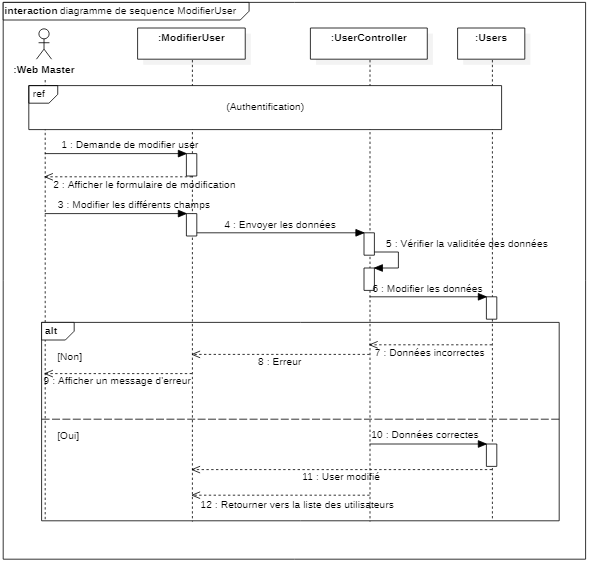
****

Figure 24: diagramme de séquence détaillé du cas d’utilisation « Modifier Utilisateur »

#### Raffinement du cas d’utilisation « Gérer profil »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Gérer profil ».

Pour gérer son profil, l’admin doit remplir le formulaire c’est-à-dire la saisie de l’ancien mot de passe, le nouveau mot de passe et la confirmation de nouveau mot de passe. Puis clique sur le bouton « Changer mot de passe » afin que le système valide les données, dans le cas où les champs entrés sont valides.

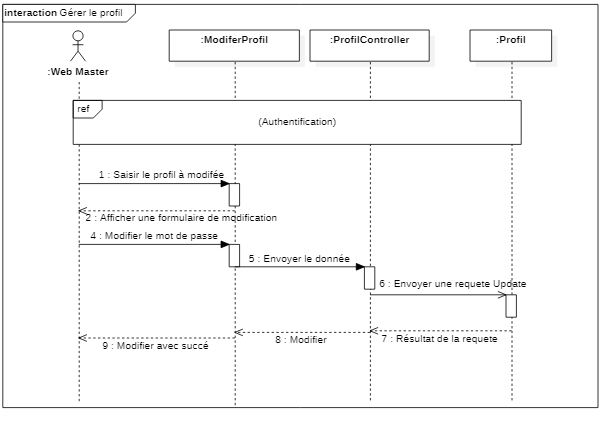


Figure 24: diagramme de séquence détaillé du cas d’utilisation « Gérer profil »

#### Raffinement du cas d’utilisation « Gérer maintenance »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Gérer la maintenance ».

Figure 25: diagramme de séquence détaillé du cas d’utilisation « Gérer maintenance »

#### Raffinement du cas d’utilisation « Gérer concours master»

* **Diagramme de séquence du CU« modifier concours »**

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Modifier concours ».

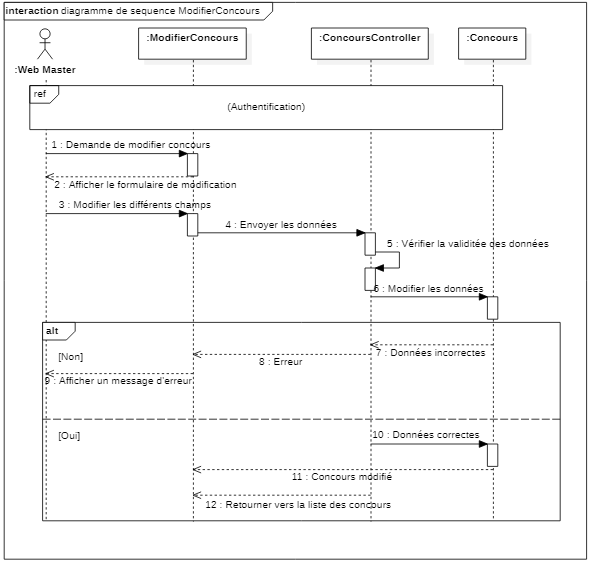


Figure 26: diagramme de séquence détaillé du cas d’utilisation « Modifier concours master »

* **Diagramme de séquence du CU« Supprimer concours »**

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Supprimer concours ».

Pour supprimer un rendez-vous. Le web master doit confirmer la suppression en cliquant sur le bouton « supprimer »

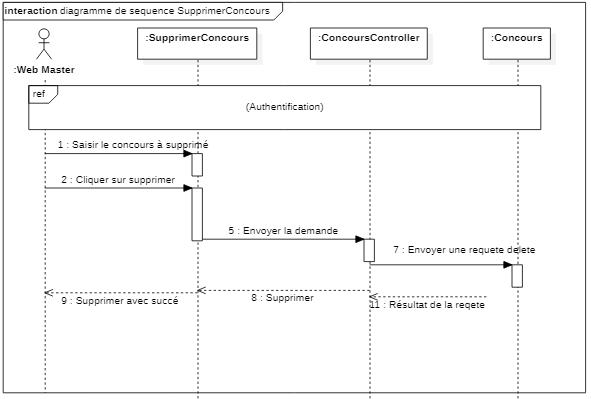


Figure 26: diagramme de séquence détaillé du cas d’utilisation « Supprimer Concours »

1. **Raffinement du cas d’utilisation « Gérer les classes »**

* **Diagramme de séquence du CU« Modifier classe »**

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Modifier classe ».

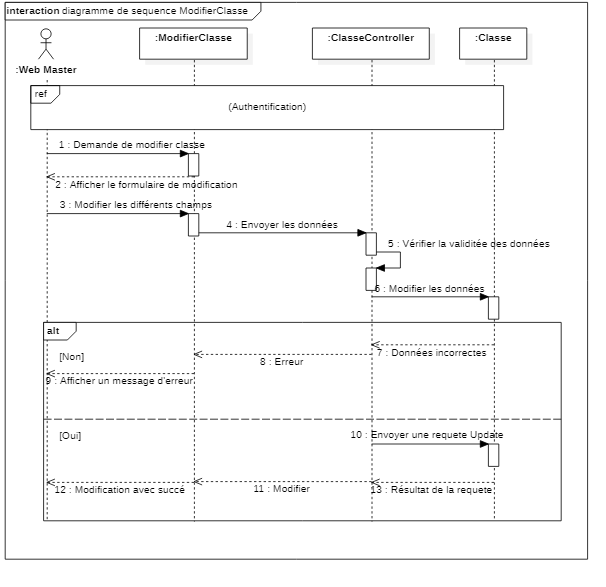


Figure 26: diagramme de séquence détaillé du cas d’utilisation « Modifier Classe »

* **Diagramme de séquence du CU« supprimer classe »**

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Supprimer classe ».

Pour supprimer un rendez-vous. Le web master doit confirmer la suppression en cliquant sur le bouton « supprimer »

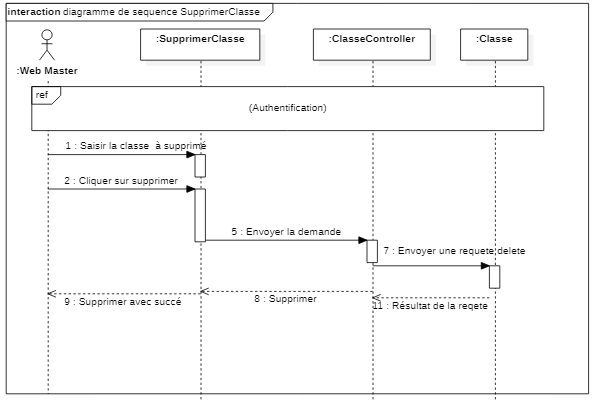


Figure 26: diagramme de séquence détaillé du cas d’utilisation «Supprimer Classe»

1. **Raffinement du cas d’utilisation « Gérer les matières »**

#### Diagramme de séquence du CU« Ajouter matière »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Ajouter matières ».

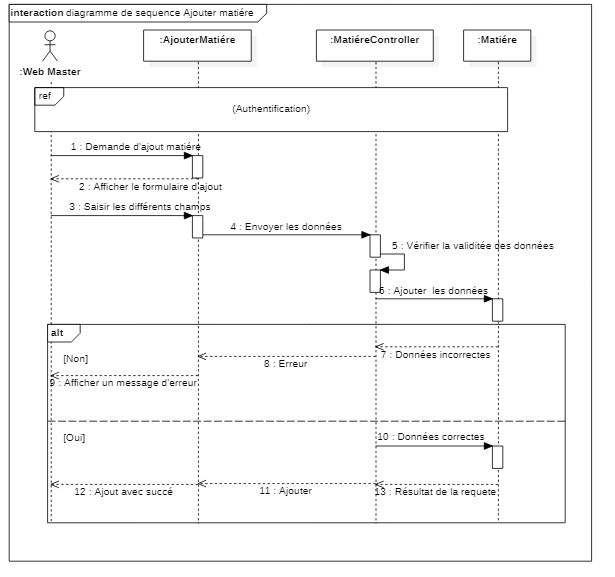


Figure 26: diagramme de séquence détaillé du cas d’utilisation «Ajouter matière»

#### Diagramme de séquence du CU« Afficher matière »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Afficher matière ».

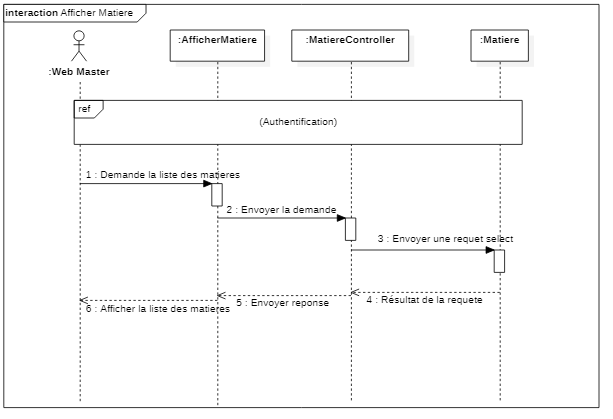


Figure 26: diagramme de séquence détaillé du cas d’utilisation «Afficher matière »

#### Raffinement du cas d’utilisation « consulter les fiches des notes et les emplois du temps »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Afficher les fiches».

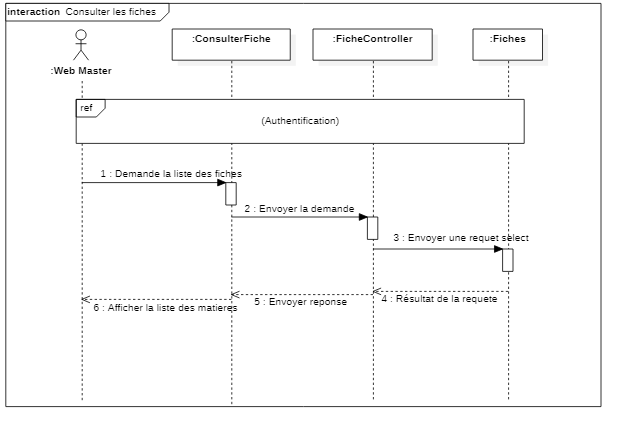


Figure 27: diagramme de séquence détaillé du cas d’utilisation « consulter les fiches »

#### Raffinement du cas d’utilisation « consulter la bibliothèque »

La figure ci-dessous représente le diagramme de séquence de conception du cas d’utilisation « Consulter la bibliothèque ».

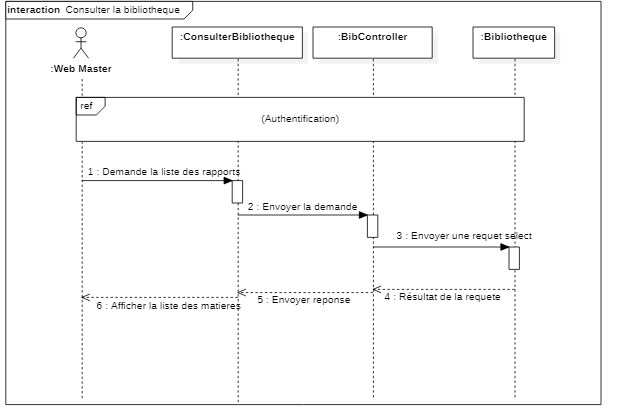


Figure 28: diagramme de séquence détaillé du cas d’utilisation « consulter bibliothèque »

#### Raffinement du cas d’utilisation « consulter les événements »

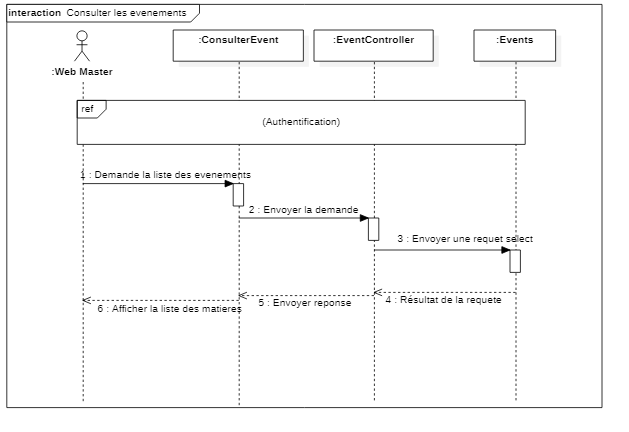


Figure 29: diagramme de séquence détaillé du cas d’utilisation « consulter les événements »

#### Raffinement du cas d’utilisation « consulter les documents administratifs »

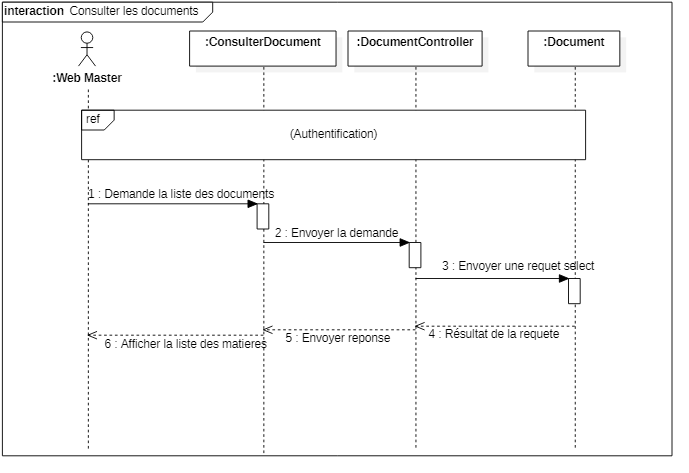


Figure 30: diagramme de séquence détaillé du cas d’utilisation « consulter les documents administratifs »

### Diagramme de classe :

Le diagramme de classe est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. Il décrit la structure de système à travers les classes, les attributs et les relations qui les associent. Le diagramme de Classe retenu à la fin de la conception est le suivant :

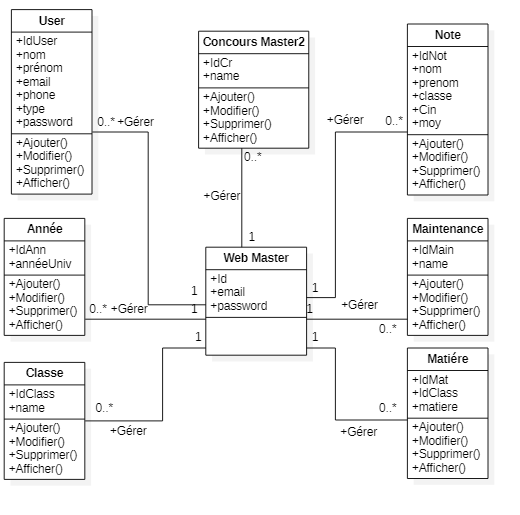


Figure 31: Diagramme de classe du sprint 1

### Description du modèle de données

Après avoir analysé le diagramme de cas d'utilisation, nous devons créer le modèle utilisateur pour commencer en cours d'exécution sur les services Web et obtenir la gestion des utilisateurs.

Dans le tableau ci-dessous (Tableau 10), nous allons donner un bref aperçu en décrivant le modèle utilisateur collection de notre application

|  |  |  |  |
| --- | --- | --- | --- |
| **Classes** | **Attributs** | **Type** | **Description** |
| Web Master | ID | Integer | Clé primaire MySQL par défaut |
| Email | String | L'email de notre utilisateur dans l'application |
| Password | String | Le mot de passe de connexion de l'application |
| User | IdUser | Integer | Clé primaire MySQL par défaut |
| Nom, Prenom | String | Le nom et le prenom de l'utilisateur dans l'application |
| Phone | Integer | Téléphone mobile de notre utilisateur dans l'application |
| Type | String | Type de l’utilisateur utilisé notre application |
| Matière | IdMat | Integer | Clé primaire MySQL par défaut |
| IdClass | Integer | Clé primaire de la classe ‘Classe’ et clé étranger de la classe ‘Matière’ |
| Matière | String |  |

Tableau : description du modèle utilisateur

### Interfaces de l’application