

Cahier des charges

de l'application

Shrimp

Version 1.0

Préparé par

CHAUVEROCHE Maxime	1	maxime.chauveroch@telecomnancy.net
COARD Alexi	2	alexi.coard@telecomnancy.net
RIGAL Nicolas	3	nicolas.rigal@telecomnancy.net
YOUSSEF Anthony	4	anthony.youssef@telecomnancy.net

Numéro de groupe : 1

Enseignant:	CHAROY François
Module:	MOCI
Groupe:	1
Chargé de TD:	GAUTIER Martine
Date:	13/10/2015

Contenu

RÉVISIONS

I INTRODUCTION

- 1.1 À PROPOS DE CE DOCUMENT
- 1.2 PORTÉE DU PRODUIT
- 1.3 DÉFINITIONS, ACRONYMES AND ABRÉVIATIONS
- 1.4 CONVENTION DE RÉDACTION DU DOCUMENT
- 1.5 PUBLIC CONCERNÉ ET VUE D'ENSEMBLE DE CE DOCUMENT

II DESCRIPTION GLOBALE

- 2.1 PERSPECTIVE DU PRODUIT
- 2.2 FONCTIONNALITÉ DU PRODUIT
- 2.3 UTILISATEURS
- 2.4 ENVIRONNEMENT D'EXÉCUTION
- 2.5 CONTRAINTES DE CONCEPTION ET D'IMPLÉMENTATION
- 2.6 MANUEL UTILISATEUR
- 2.7 HYPOTHÈSES ET DÉPENDANCES

III BESOINS SPÉCIFIQUES

- 3.1 BESOINS EXTERNES
 - 3.1.1 INTERFACE UTILISATEUR
 - 3.1.2 INTERFACE MATÉRIELLE
 - 3.1.3 INTERFACE LOGICIELLE
 - 3.1.4 INTERFACE DE COMMUNICATION
- 3.2 BESOINS FONCTIONNELS
- 3.3 BESOINS COMPORTEMENTAUX
 - 3.3.1 DIAGRAMME DE CAS D'UTILISATION
 - 3.3.2 DÉTAILS DES CAS D'UTILISATION

IV AUTRES BESOINS NON FONCTIONNELS

- 4.1 BESOINS LIÉS AUX PERFORMANCES
- 4.2 BESOINS LIÉS À LA SÉCURITÉ
- 4.3 BESOINS LIÉS À LA QUALITÉ DU PRODUIT
- 4.4 AUTRES BESOINS

APPENDICE A – DICTIONNAIRE DE DONNÉES

APPENDICE B - GESTION DE PROJET

Révisions

Version	Auteurs	Description de la version	Date de soumission
1.0	CHAVEROCHÉ Maxime COARD Alexi RIGAL Nicolas YOUSSEF Anthony	Première version du cahier des charges	13/10/2015

I. Introduction

1. À propos de ce document

Le but de ce document est de fournir une description en profondeur de l'ensemble des fonctionnalités de Shrimp. L'ensemble des fonctionnalités de Shrimp sera présenté, ainsi que leurs objectifs et leur fonctionnement à l'aide de descriptions textuelles et graphiques. L'environnement technique de l'application sera aussi détaillé dans ce document.

Ce document s'adresse tout d'abord au client de cette application, l'Université de Lorraine (UL) mais aussi aux futurs développeurs et utilisateurs maintenance de Shrimp.

2. Portée du produit

Le but de Shrimp est de simplifier la gestion des absences des élèves de l'UL en informatisant les relations entre les administrations, les professeurs et les étudiants. Le système laisse une grande marge de manœuvre aux étudiants sans pour autant automatiser les tâches propres aux administrations. En proposant un fonctionnement simple, le système pourra être rapidement adopté par les équipes pédagogiques qui gagneront du temps au moment des appels.

Les développeurs de Shrimp sont aussi conscients des enjeux de sécurité et de protection des données, c'est pourquoi ils intégreront les systèmes d'identifications de l'UL pour la connexion à Shrimp et la gestion des documents privés des utilisateurs.

3. Définitions, acronymes et abréviations

- API (**A**pplication **P**rogramming **I**nterface) : interface de programmation permettant à un logiciel d'offrir ses services à d'autres logiciels
- ADE : Application web de l'Université de Lorraine permettant de consulter les listes d'élèves, leurs emploi du temps etc
- CAS (**C**entral **A**uthentication **S**ervice) : service d'authentification utilisé par l'UL dans ADE.
- GHz (**G**iga**H**ertz) : unité de mesure de la vitesse du processeur
- GO (**G**iga**O**ctet) : unité de mesure équivalent à 1024 Mo
- HTTPS (**H**yper **T**ext **T**ransfert **P**rotocol **S**ecure) : protocole permettant d'établir des connexions sécurisées pour transférer des données

- Historique des activités : Comporte l'historique de toutes les actions effectuées par les utilisateurs de l'application sous forme simplifiée.
- Log: Un fichier de log est un fichier comportant l'historique de toutes les actions effectuées par une application, celui-ci est réservé aux développeurs de l'application. Il est plus détaillé que l'historique des activités.
- Mo (**M**ega**O**ctet) : unité de mesure équivalent à 1 048 576 octets
- To (**T**era**O**ctet) : unité de mesure équivalent à 1024 Go
- UL: **U**niversité de **L**orraine

4. Conventions de rédaction du document

Dans un souci d'uniformisation et de clarté lors de la rédaction de ce cahier des charges, nous avons eu recours à des notations et conventions d'écriture spécifiques.

Ce document utilise la police de caractère Arial de taille 11 dans l'intégralité des textes présents, à l'exception des titres de section qui eux, sont en taille 24 écrits en gras. Les sous-titres sont eux en taille 14.

Nous avons respecté la convention de notation des sections et titres suivante :

- Les titres de sections sont numérotés en chiffres romains (I, II, III, IV etc)
- Les titres des sous sections sont numérotés en chiffres arabes (1, 2, 3 etc)

Lorsque nous avons eu besoin de lister des éléments, nous avons utilisé des listes à puces afin de permettre au lecteur de repérer facilement chaque élément d'une énumération. Le texte est quant à lui entièrement justifié et un alinéa marque chaque début de paragraphe.

Chaque diagramme et schéma est titré, il comporte également une légende permettant la compréhension de ce dernier. Chaque acronyme ou abréviation est décrit dans la section éponyme, cette description comporte la signification de ces initiales ainsi qu'une brève description de ce dernier.

Tout commentaire relatif à une information est écrit en italique et de la même manière, toute donnée jugée importante sera en gras.

5. Public concerné et vue d'ensemble de ce document

Le reste de ce document est divisé en 3 sous-parties : une description globale de Shrimp, une description des besoins spécifiques, une description des besoins non fonctionnels.

Dans la section “Description globale” seront présentées les fonctionnalités générales de Shrimp à l’aide de quelques schémas afin de faciliter la compréhension des différents objectifs de l’application. Cette partie est destinée en priorité aux futurs clients de Shrimp et ne contient que peu de détails techniques.

À l’inverse, la section “Besoins spécifiques” qui suit, est principalement à destination des développeurs puisqu’elle contient les spécifications rédigées avec des termes techniques ainsi que des diagrammes de fonctionnement (diagrammes de séquences, diagrammes d’activité et cas d’utilisation).

La section suivante, “Autres besoins non fonctionnels”, est une section vitale et cela, autant pour les clients et que les développeurs que pour le(s) futur(s) administrateur(s) de l’application. D’une part parce que certains besoins non fonctionnels sont ajoutés par l’équipe de développement et ont donc besoin d’être discuté avec le client, et d’autre part parce que ces besoins, moins rigides que les besoins fonctionnels, peuvent être modifiés pendant le développement de Shrimp. Les besoins non fonctionnels concernent les performances, la sécurité et la qualité de Shrimp mais aussi les besoins divers qui ne rentrent pas dans l’une de ces 3 catégories.

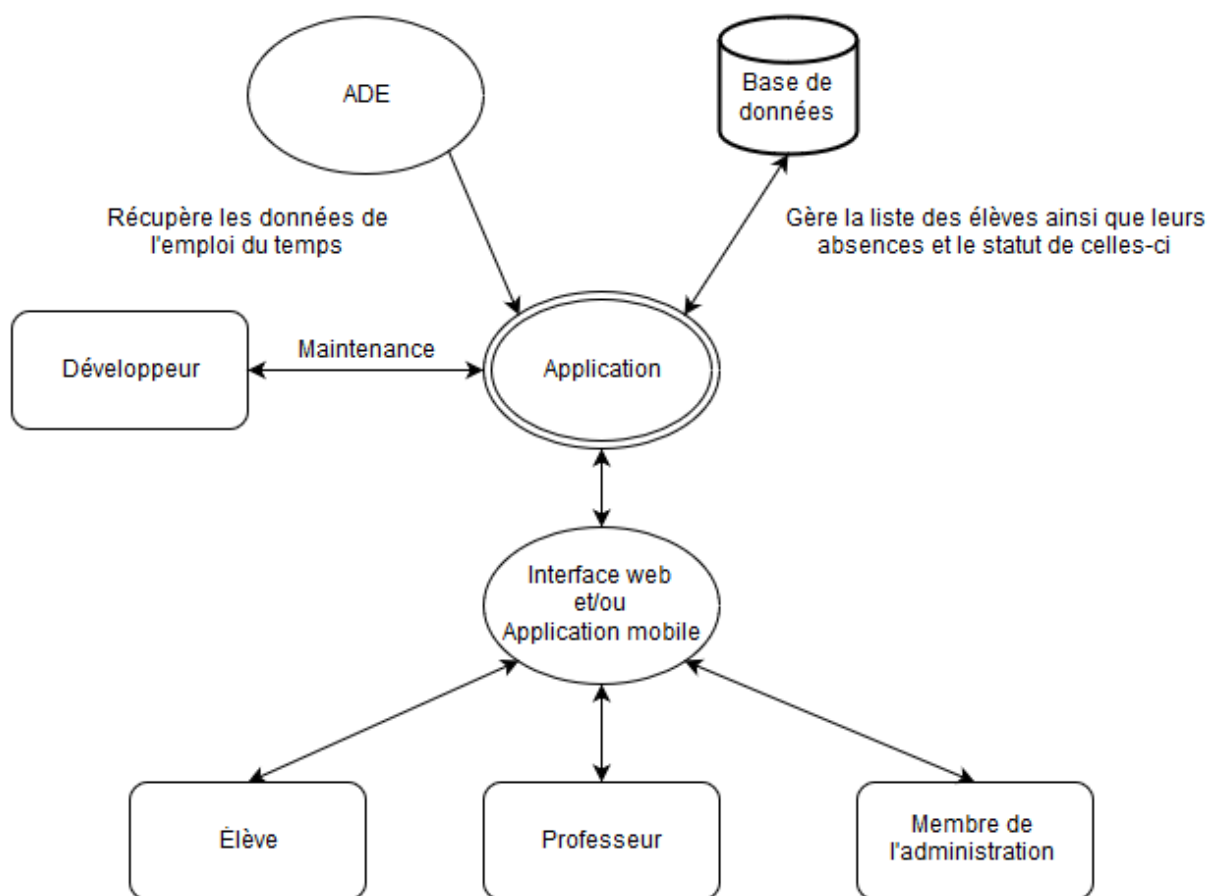
II. Description globale

1. Perspective du produit

Le besoin originel de ce produit provient de l'absence de système de gestion des absences automatisé au sein de l'Université de Lorraine. Jusqu'ici, l'appel était fait sur papier par les professeurs, puis une saisie sur Excel était effectuée par la scolarité. Les élèves ne pouvaient alors consulter leurs absences que par le biais de feuilles Excel exportées au format PDF tous les 3 mois.

Les acteurs "professeur" et "scolarité" seront repris en compte dans ce produit, et les acteurs "étudiants" et "maintenance" seront ajoutés. Le produit utilisera de plus la plate-forme ADE déjà en place dans l'infrastructure de l'Université de Lorraine afin d'en récupérer l'emploi du temps et de le modifier.

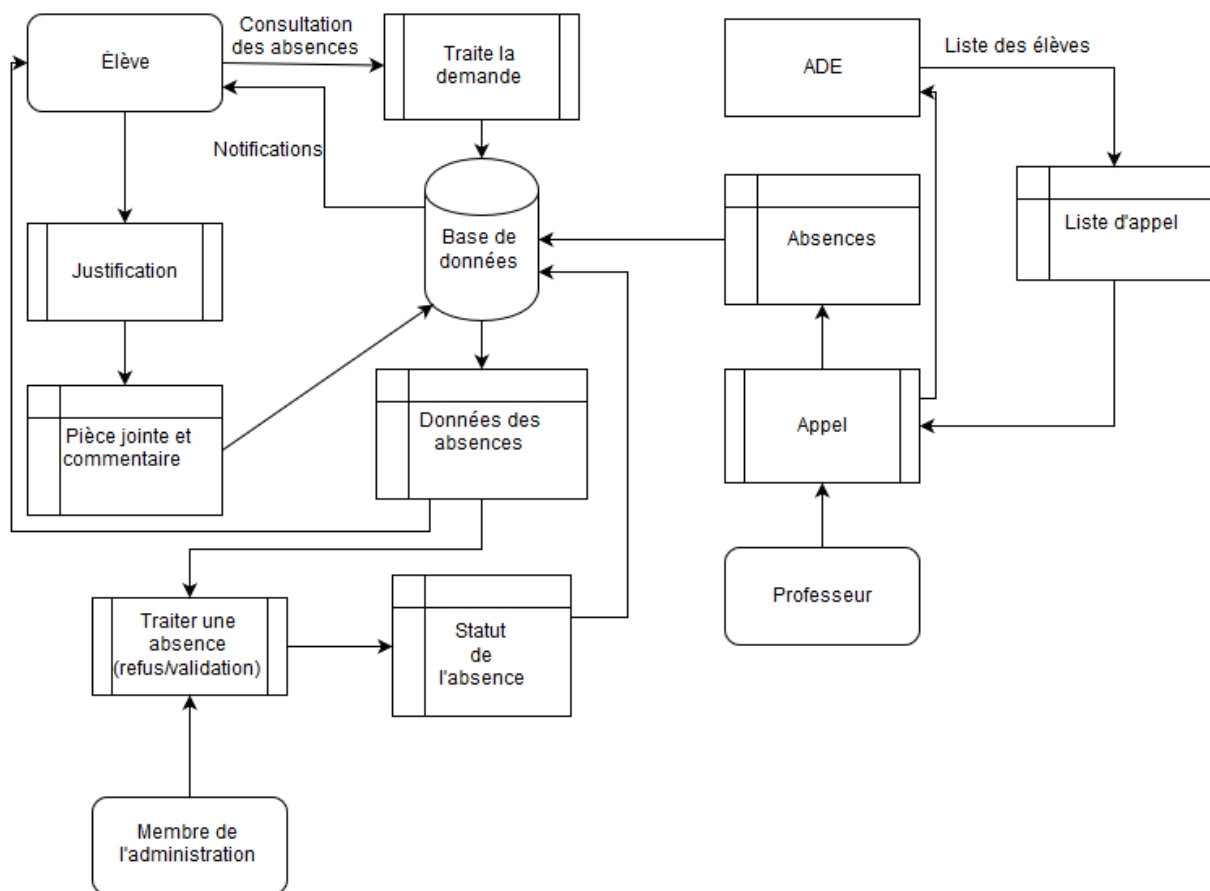
Contexte et interactions entre l'application et l'environnement



2. Fonctionnalités du produit

Il s'agit pour ce produit de permettre aux étudiants de consulter leurs absences en temps réel et de les justifier, de permettre aux professeurs de saisir les absences au moment de l'appel et de permettre à la scolarité de valider (ou refuser) les justifications des élèves. L'application devra aussi générer des fichiers logs afin de conserver une trace de chaque action et erreur survenue.

Diagramme de flot de données de l'application



3. Utilisateurs

Étudiant :

- fixe/mobile
- utilise consultation d'absences et justification
- confronté occasionnellement au système de gestion des absences
- compétence technique moyenne

Scolarité :

- fixe
- utilise saisie, consultation et validation d'absences
- confrontée régulièrement au système de gestion des absences
- compétence technique faible

Professeur :

- mobile
- utilise saisie et consultation d'absences
- confronté régulièrement au système de gestion des absences
- compétence technique faible

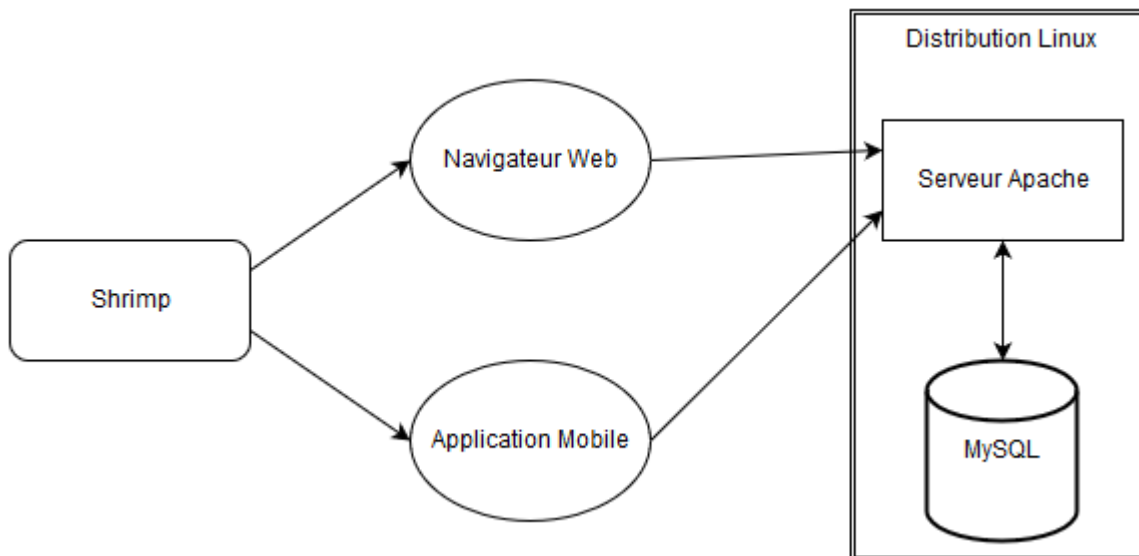
Maintenance :

- fixe/mobile
- utilise les fichiers logs
- confronté occasionnellement au système de gestion des absences
- compétence technique élevée

Considérant la réunion de la fréquence d'utilisation la plus élevée et la compétence technique la plus basse, cet ensemble de critères pointe vers les utilisateurs Scolarité et Professeur. Ce sont ces utilisateurs qui nécessiteront le plus d'attention (fréquence) et le plus de travail concernant la facilité d'utilisation (compétence technique faible). Par ailleurs, ces deux utilisateurs n'utilisent que les fonctionnalités "saisie, consultation et validation d'absences". Ces trois fonctionnalités constituent donc le coeur de l'application.

4. Environnement d'exécution

Diagramme d'environnement de l'application



Le produit s'exécutera par l'intermédiaire du serveur web Apache sur un serveur matériel doté d'une distribution Linux dans une version minimale. Ce serveur sera situé dans l'infrastructure de l'Université de Lorraine (le choix de l'emplacement du serveur est laissé au client), et ne sera accessible aux utilisateurs que par l'intermédiaire d'un navigateur web. L'application sera interfacée avec le système de gestion de base de données relationnel MySQL, présent lui aussi sur le serveur. Les ordinateurs côté utilisateur devront comprendre un navigateur web Internet Explorer 7 et supérieur y compris sa version Windows 10 (Edge), Mozilla Firefox, Google Chrome, Safari ou Opera, ainsi que d'une mémoire vive supérieure ou égale à 2Go afin de pouvoir accéder à l'application.

L'application mobile devra être compatible et fluide sur la plupart des smartphones actuels, ce qui implique des téléphones ayant au minimum un processeur dual core à 1.2GHz.

5. Contraintes de conception et d'implémentation

Le produit devra utiliser l'application ADE par l'intermédiaire de son API web afin d'y modifier les emplois du temps et de les consulter. L'application web devra être responsive et s'adapter à un écran pouvant aller d'une résolution de 320x480 pixels à 1920x1080 pixels. De plus, l'application devra être "légère", i.e. consommer le moins de ressources possible (principalement en termes de flux réseau et de RAM). En particulier, l'application devra utiliser la technologie AJAX. L'application utilisera le système de gestion de base de données relationnel MySQL pour le stockage des identifiants, des absences et des justificatifs. Pour des raisons de maintenance et d'efficacité des requêtes HTTP, l'application devra être RESTful.

6. Manuel utilisateur

Un court tutoriel (quick-start) sera déclenché dès la première ouverture de session d'un utilisateur afin de le guider dans l'interprétation de l'interface graphique ainsi que des fonctionnalités à sa portée. La durée du tutoriel variera suivant le type d'utilisateur, les étudiants et les professeurs ayant moins d'éléments de l'application à maîtriser que les membres des administrations. Ce "quick-start" pourra être rejoué à tout moment depuis l'interface graphique. Techniquement, ce tutoriel sera sous format d'une fonction JavaScript.

7. Hypothèses et dépendances

Hypothèses de conception :

- Accès à l'API du logiciel ADE
- Autorisation d'accès en modification des emplois du temps dans la base de données de ADE
- Accès à l'API de CAS pour la gestion des authentification des utilisateurs UL
- L'UL a les moyens physiques et financiers d'installer un serveur (dont les performances ont été décrites plus bas dans la partie "Besoins non-fonctionnels") dans son infrastructure
- L'UL a les moyens de sécuriser physiquement ce serveur

III. Besoins spécifiques

1. Besoins fonctionnels

L'application doit avoir différentes fonctionnalités que nous avons classé en différentes catégories.

Saisie des absences

L'application doit permettre de saisir les absences des élèves. Pour cela, les professeurs ont à leur disposition l'application (sous son format web ou mobile) leur proposant une interface afin de réaliser cette action.

Dans ce type de besoin, l'application doit fournir les fonctionnalités suivantes :

- un professeur peut faire l'appel pendant son cours
- la scolarité peut valider cet appel

Justification des absences

Les élèves peuvent justifier leur absence après que celle-ci ai été validée. Pour cela, ils peuvent le faire de deux manières. Ils peuvent soit aller fournir un justificatif en format papier à l'administration, soit utiliser l'interface Shrimp afin de faire une demande de justification. Ce demande est alors traitée par la scolarité qui valide ou non l'absence.

Dans cette catégorie, l'application doit fournir les fonctionnalités suivantes :

- un élève peut justifier son absence
- la scolarité peut traiter une absence en l'acceptant ou en la refusant
- la scolarité peut justifier l'absence d'un élève

Consultation des absences

Chaque élève peut consulter ses absences, de même un professeur peut consulter les absences de ses élèves. La scolarité quant à elle a accès aux absences de tout les élèves.

L'application doit fournir les fonctionnalités suivantes

- un élève peut consulter ses absences
- un professeur peut consulter les absences de ses élèves
- la scolarité peut consulter les absences de n'importe quel élève.

Absence des professeurs

Lorsqu'un professeur prévient de son absence pour un certain cours, la scolarité peut annuler le cours du professeur en question ce qui prévient par la suite le professeur et les élèves concernés L'application doit permettre à la scolarité d'annuler le cours d'un professeur.

Notifications

Lorsqu'un cours est annulé, ou qu'une absence est signalée ou traitée par la scolarité, l'application envoie une notification aux personnes concernées. Cette notification apparaîtra selon le choix de l'utilisateur sur son téléphone mobile ou sur l'onglet notification de l'application.

L'application doit fournir les fonctionnalités suivantes:

- Informer un élève qu'une absence a été notifiée
- Informer un élève que son absence a été traitée (validée ou refusée)
- Informer un professeur et les élèves concernés qu'un cours est annulé

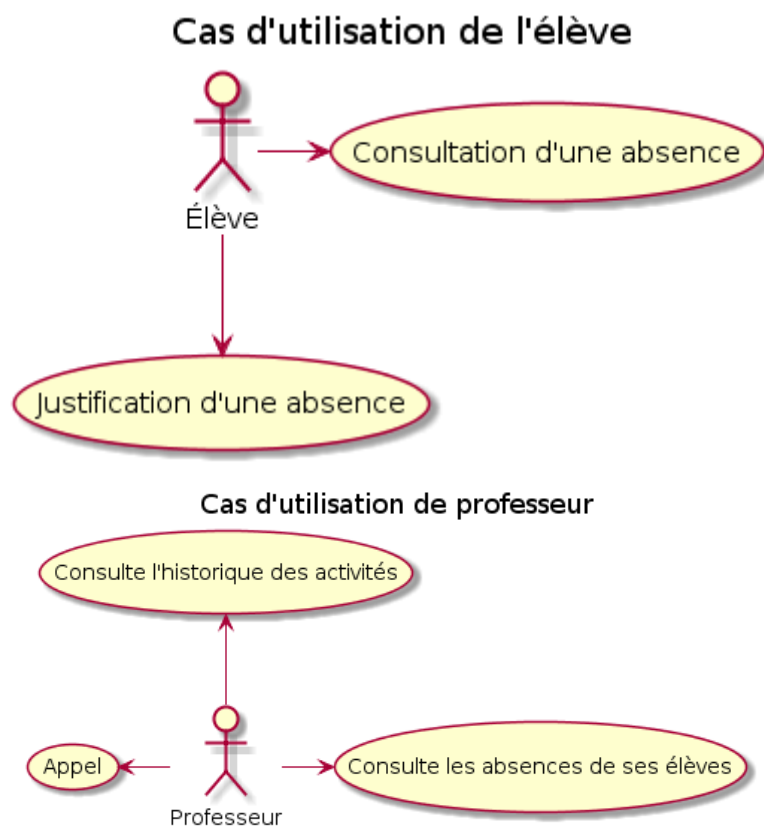
Fichier de log

Afin de faciliter la maintenance de l'application, deux fichiers de logs seront générés, un uniquement pour le développeur chargé de la maintenance de l'application, et un plus concis pour la scolarité.

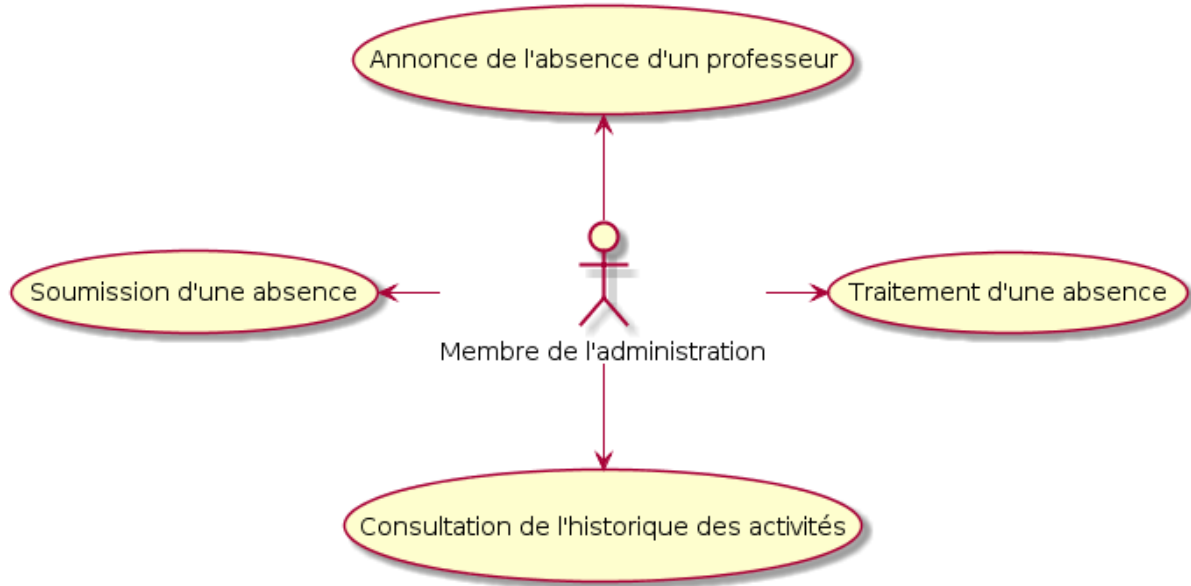
L'application doit donc remplir les conditions suivantes:

- Générer un fichier de logs concis pour la scolarité et autoriser son accès à cette dernière
- Générer un fichier de logs détaillé accessible uniquement au développeur

2. Diagramme de cas d'utilisation



Cas d'utilisations d'un membre de l'administration

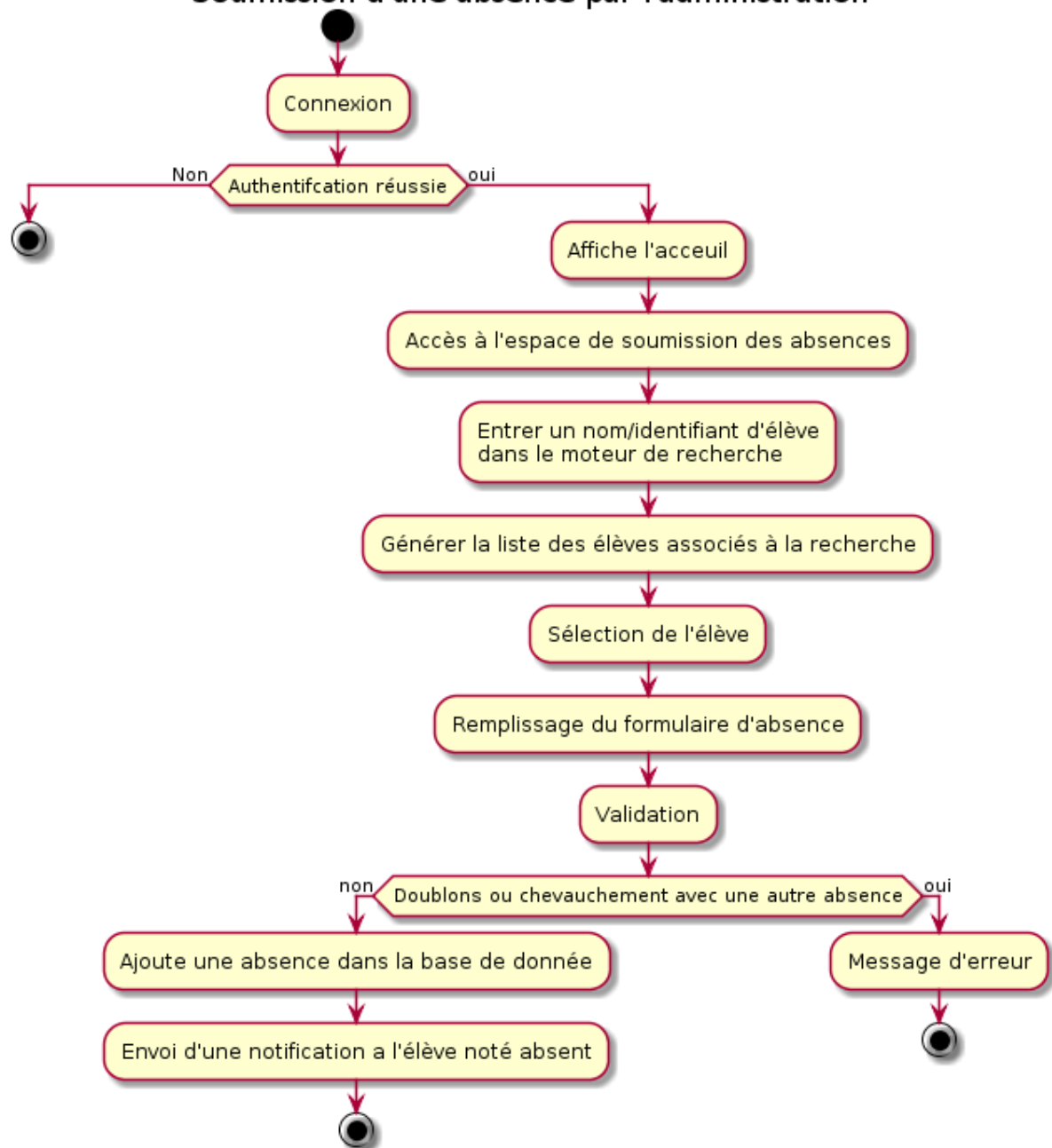


Cas d'utilisations d'un administrateur

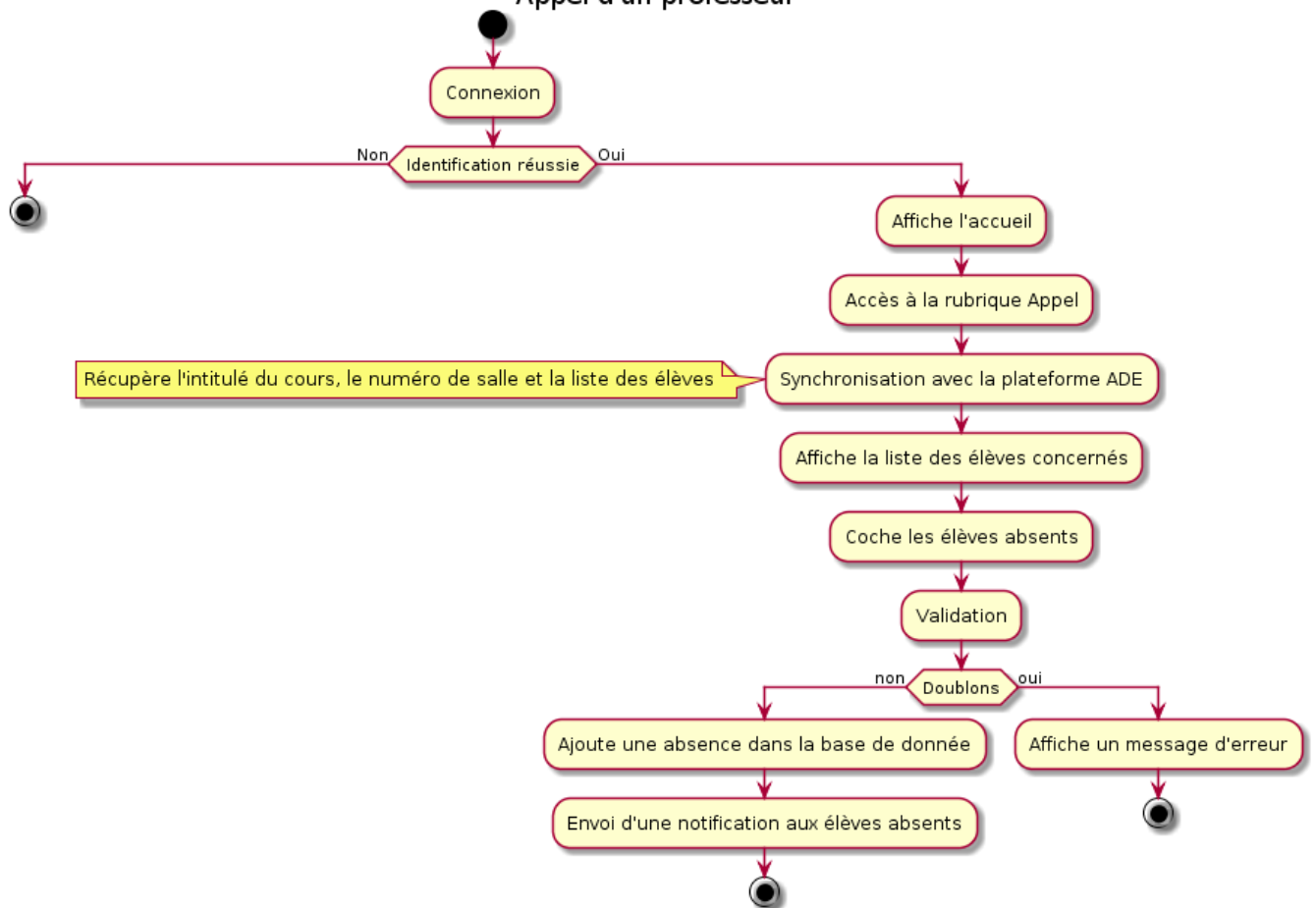


3. Détails des cas d'utilisation

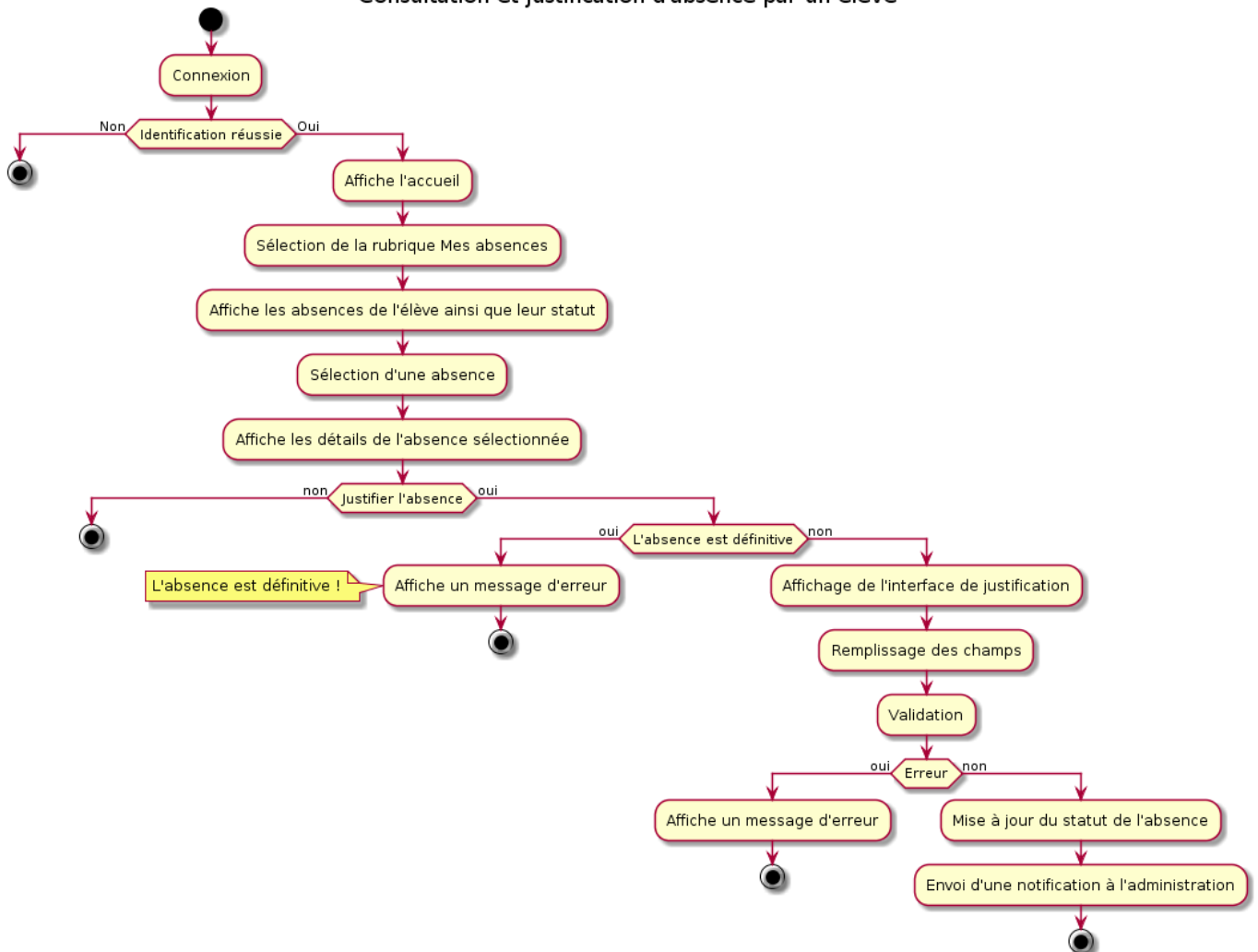
Soumission d'une absence par l'administration



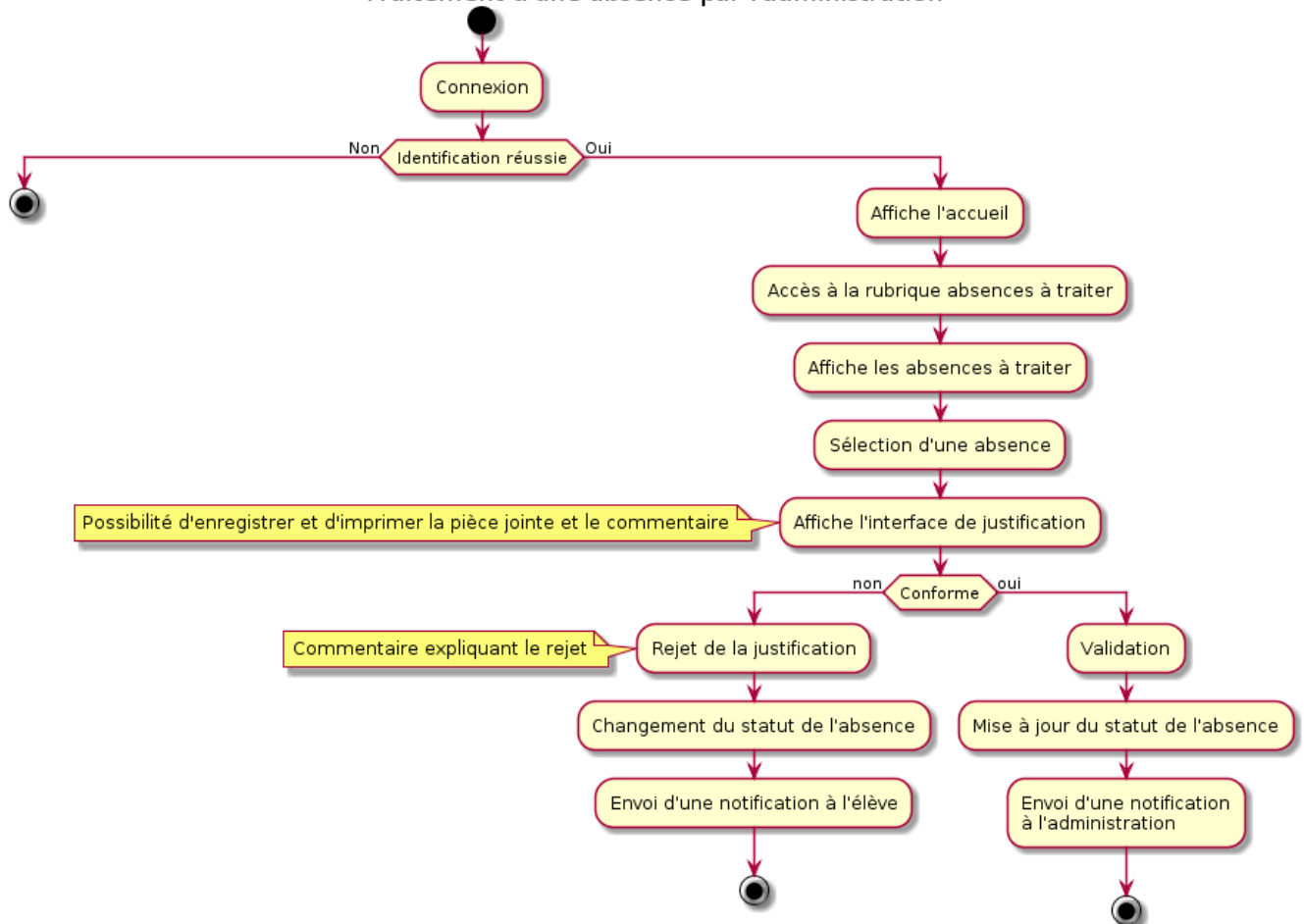
Appel d'un professeur



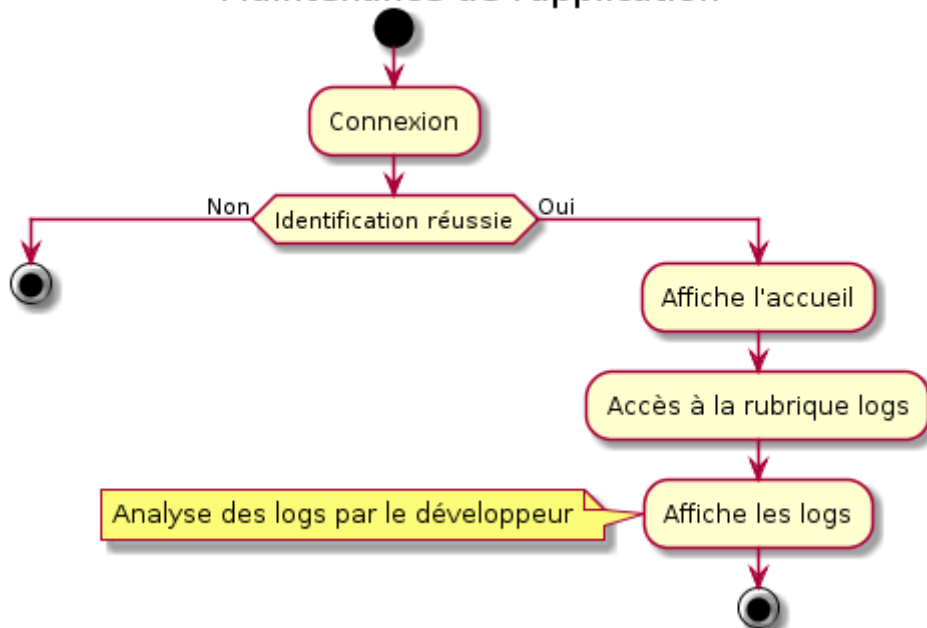
Consultation et justification d'absence par un élève



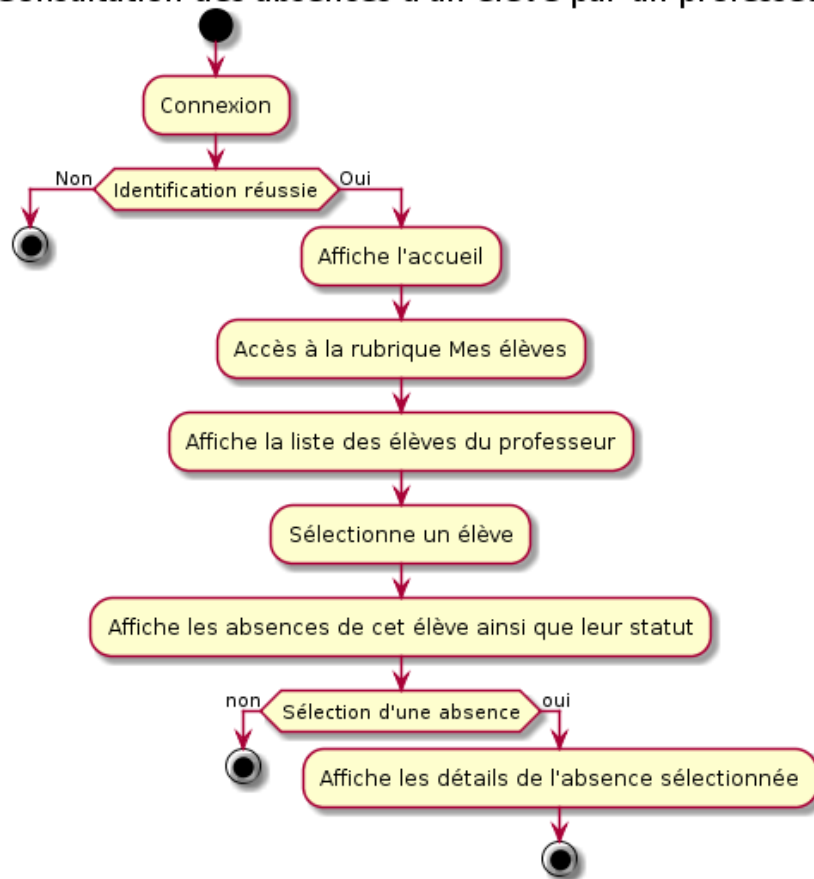
Traitement d'une absence par l'administration



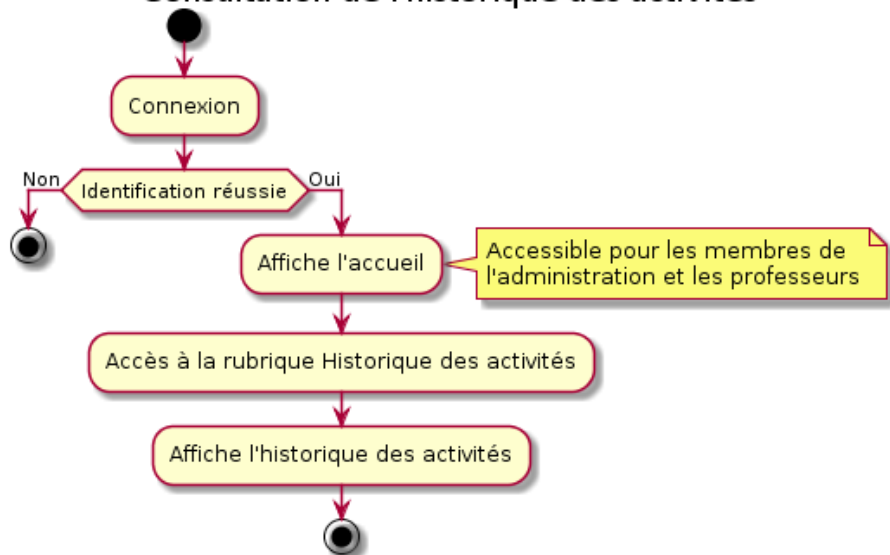
Maintenance de l'application



Consultation des absences d'un élève par un professeur



Consultation de l'historique des activités



4. Interfaces utilisateur

Commun à tous les utilisateurs

Il y aura un écran de connexion pour navigateur web et un autre écran de connexion spécialement pour les appareils mobiles. Suite à la connexion, l'accueil prendra la forme correspondant au statut de la personne se connectant (Élève, Professeur ou Membre de l'administration) et permettra d'accéder aux écrans concernant les différents usagers.

De plus, un bandeau indiquant les tâches urgentes (absence à justifier, absence sur le point de devenir définitive pour les élèves ; absences à traiter pour les membres de l'administration et autres) pourra apparaître en haut de l'écran d'accueil pour une notification immédiate des usagers.

Suite à la connexion, un lien "Déconnexion" sera visible à tout moment dans le coin supérieur droit et permettra la déconnexion rapide de l'utilisateur s'il vient à l'utiliser. Une fois dans un écran autre que l'écran d'accueil, un bouton "Retour" apparaîtra et sera visible à tout moment dans le coin supérieur gauche. Ce bouton permettra de revenir dans l'écran d'accueil.

Partie Élèves

- Pour l'écran d'accueil de l'élève, celui-ci permettra d'accéder à l'écran de consultation de l'emploi du temps et à celui des absences.
- L'écran d'emploi du temps affichera l'emploi du temps de l'élève sous la même forme le fait ADE.
- L'écran des absences affichera la liste des absences de l'élève ainsi que leur statut. Un bouton "Détails" apparaîtra à la droite de chaque absence. Ce bouton permet d'obtenir l'écran "Détail d'une absence" qui affichera toutes les précisions concernant l'absence en question, ainsi que le bouton de justification. Ce bouton mène à l'écran de justification qui permet de fournir le chemin de la pièce jointe (un bouton "Explorer" permet de faire cela via l'explorateur de fichiers de l'OS) et également de fournir des commentaires si nécessaire.

Partie Professeurs

- L'écran d'accueil du professeur quant à lui permet d'accéder aux écrans d'appel, de consultation d'emploi du temps, de consultation des absences des élèves ainsi qu'à l'écran de signalement de son absence.
- L'écran d'appel affichera directement le nom du cours actuel, ses horaires de début et fin ainsi que la liste des élèves censés assister au cours (chaque nom est précédé d'une case pouvant être cochée), le professeur n'a plus qu'à cocher chaque élève présent puis à cliquer sur le bouton "Valider".
- L'écran d'emploi du temps affichera l'emploi du temps de l'élève sous la même forme le fait ADE.

- L'écran de consultation des absences des élèves affichera la liste des élèves du professeur ainsi que le nombre d'absences injustifiées de ces élèves. Un bouton "Détails" sera à la droite de chaque élève ; ce bouton affichera le détails des absences de l'élève concerné.
- L'écran de signalement de son absence permettra d'accéder à un formulaire de signalement d'absence qui contiendra le jour et l'heure de début et fin d'absence ainsi qu'une zone réservée aux commentaires facultatifs.

Partie Membres de l'administration

- L'écran d'accueil des membres de l'administration permettra d'accéder à l'écran de soumission d'une absence et à celui de traitement des absences.
- L'écran de soumission d'une absence permettra de soumettre au système l'absence d'un élève. Pour cela, il affichera la liste des élèves rattachés à l'administration, chacun sera accompagné d'un bouton "Signaler l'absence". Une fonction de recherche sera également présente. Une fois l'élève trouvé, le bouton "Signaler l'absence" fais apparaître dans un cadre le formulaire de signalement de l'absence d'un élève qui comprend une date et une heure de début et de fin d'absence, ainsi qu'une zone de commentaires.
- L'écran de traitement des absences affichera la liste des absences à traiter ainsi que leur statut. Un bouton "Détails" sera affiché à la droite de chaque absence et permettra de faire apparaître dans un cadre les détails d'une absence : ceux-ci comprendront le nom et prénom de l'élève ainsi que la période de l'absence, la pièce justificative fournie par l'élève accessible via le bouton "Pièce justificative" et le commentaire s'il y a lieu. Les boutons "Refuser la justification" et "Accepter la justification" apparaîtront en bas du cadre.

Connexion via Navigateur Web

The diagram illustrates a web browser window with a grey header bar. Inside the header, the text "Connexion sur Shrimp" is centered. To the left of the text are four navigation icons: a left arrow, a right arrow, a close 'X' button, and a home icon. To the right is a search bar containing the URL "http://shrimp.univ-lorraine.com" and a magnifying glass icon. The main content area is white and contains the text "SHRIMP" at the top. Below this, there are two labels: "Identifiant UL" and "Mot de passe", each followed by a rectangular input field. The browser window has a grey footer bar with a small double-slash icon in the bottom right corner.

Connexion sur Shrimp

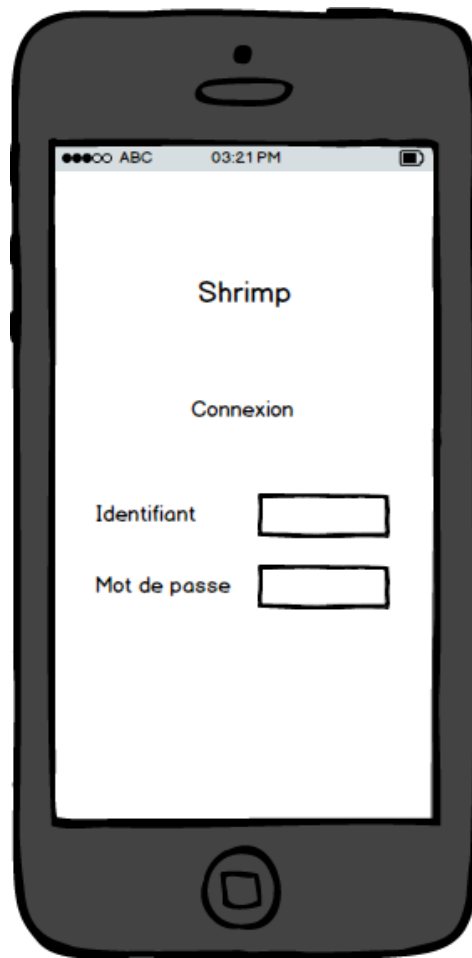
<http://shrimp.univ-lorraine.com>

SHRIMP

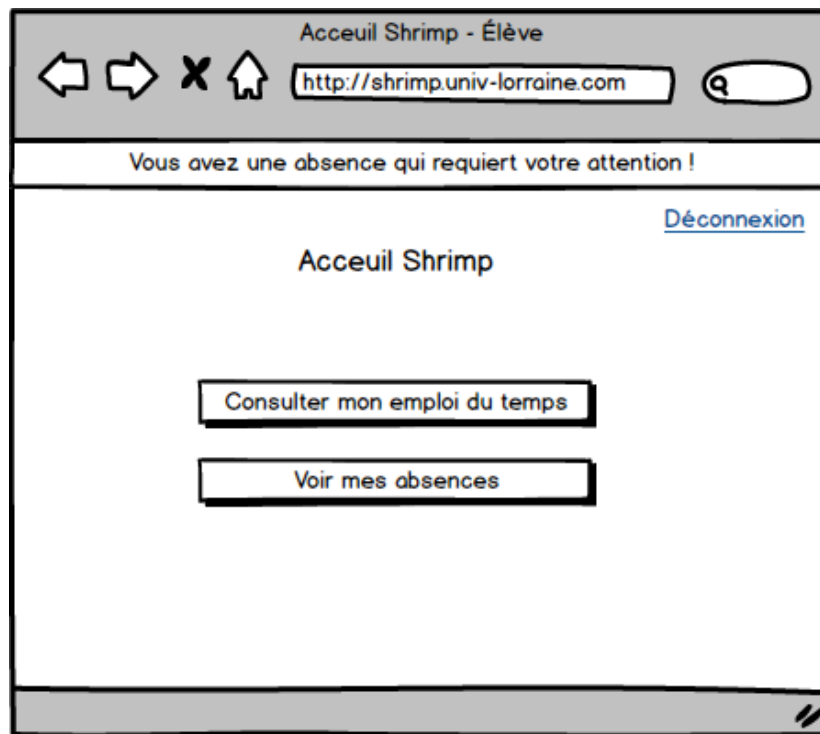
Identifiant UL

Mot de passe

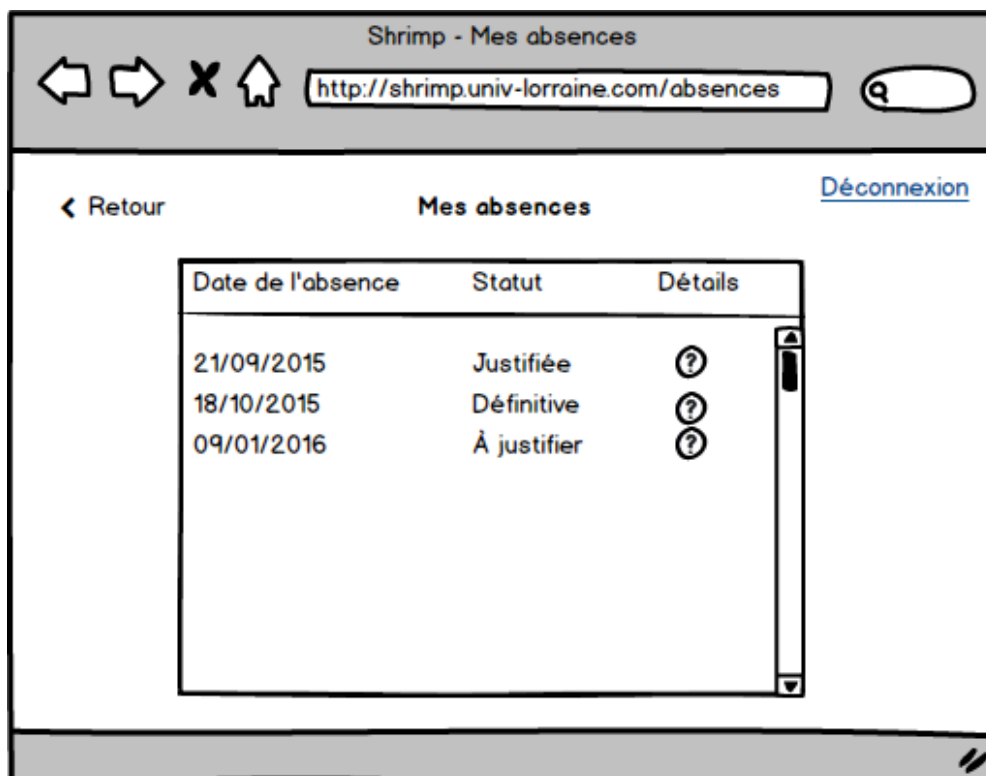
Connexion via Mobile



Accueil - Élève





Vue absences - Élève



Détail d'une absence - Élève


Shrimp - Mes absences



[< Retour](#)



Mes absences

[Déconnexion](#)

Date de l'absence	Statut	Détails
Absence du 9/01/2016		
Statut :		À justifier
Cours manqués :		
<div>Justifier cette absence</div>		

Justifier une absence - Élève

Shrimp - Mes absences



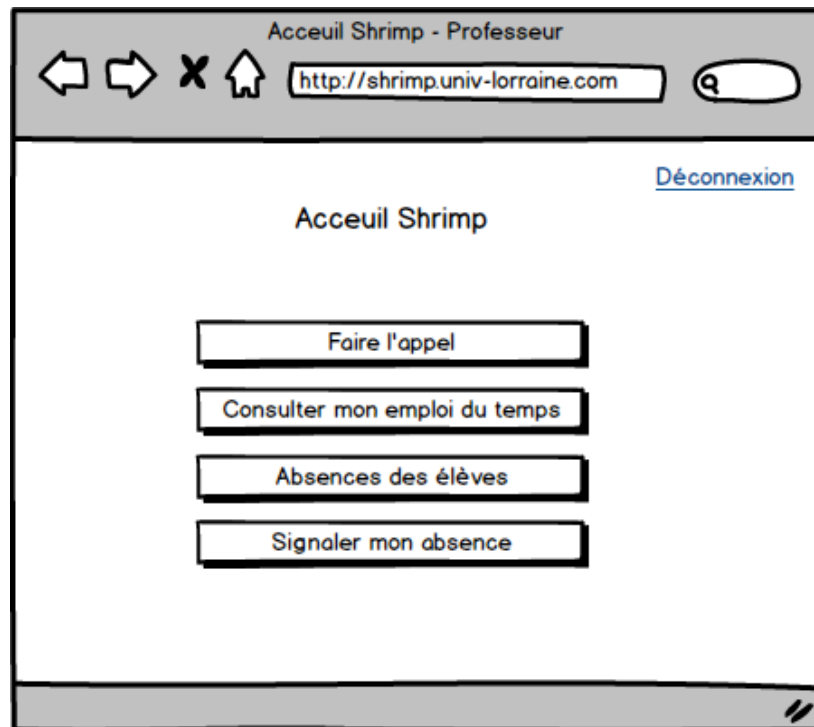
[< Retour](#)

Mes absences

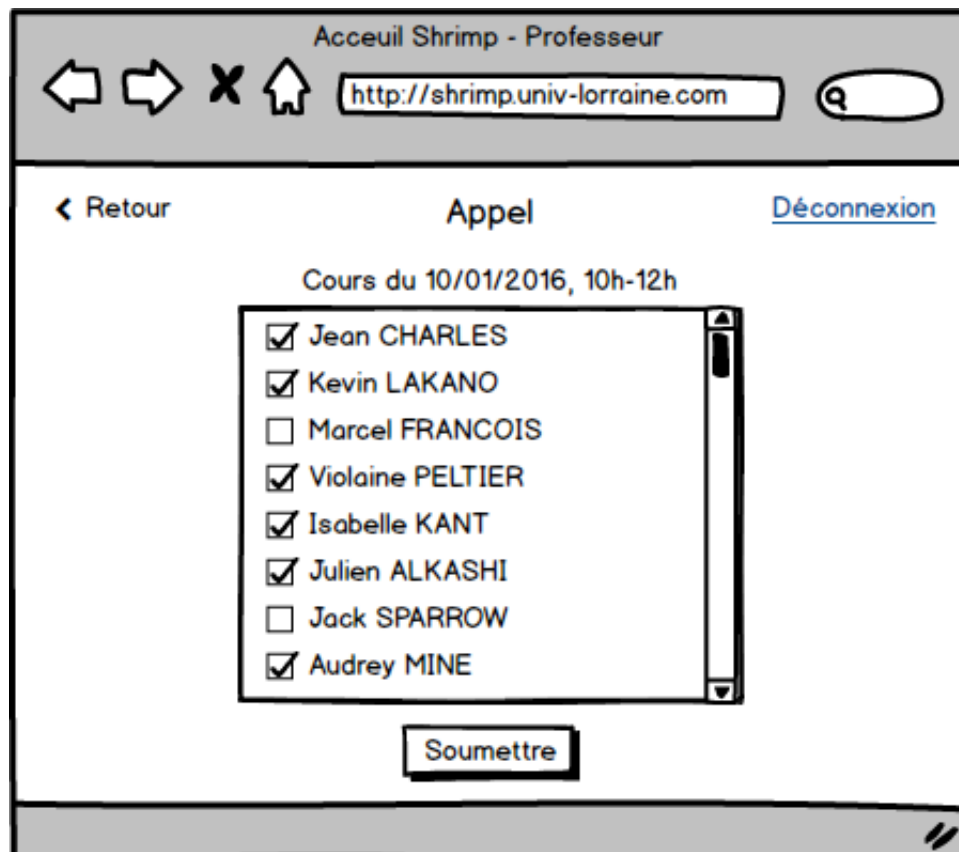
[Déconnexion](#)

Date de l'absence	Statut	Détails
Justification de l'absence du 9/01/2016		
Chemin d'accès vers la pièce justificative		
<input type="text"/>		<div>Parcourir</div>
Commentaires (facultatif)		
<input type="text"/>		
<div>Soumettre</div>		

Accueil - Professeur



Appel - Professeur



Consultation des absences des élèves - Professeur

Shrimp - Absences de mes élèves

← → ✕ 🏠 🔍

← Retour **Mes élèves** [Déconnexion](#)

Nom et prénom	Nombre d'absences injustifiées	Détails
NEMAR Jean	0	?
MARCHECIEL Luc	2	?
TOUSEUL Yan	1	?
BAMBEL Larry	9	?

Signaler son absence - Professeur

Accueil Shrimp - Professeur

← → ✕ 🏠 🔍

← Retour **Signaler mon absence** [Déconnexion](#)

Du 📅 à h

au 📅 à h

Commentaires :

Accueil - Membre de l'administration

The image is a hand-drawn sketch of a web browser window. The title bar at the top reads "Accueil Shrimp - Membre de l'administration". Below the title bar is a navigation bar containing back, forward, and home icons, a search icon, and a text input field with the URL "http://shrimp.univ-lorraine.com". Below the navigation bar is a white banner with the text "Vous avez une absence à traiter !". To the right of the banner is a blue link labeled "Déconnexion". The main content area has the heading "Accueil Shrimp" and two buttons: "Soumettre une absence" and "Traiter les absences". The browser window has a grey border and a small icon in the bottom right corner.

Accueil Shrimp - Membre de l'administration

Vous avez une absence à traiter !

[Déconnexion](#)

Accueil Shrimp

Soumettre une absence

Traiter les absences

Shrimp - Soumettre l'absence d'un élève

http://shrimp.univ-lorraine.com/absences

< Retour

Rechercher OK

Liste des élèves

10/10

Shrimp - Soumettre l'absence d'un élève

http://shrimp.univ-lorraine.com/absences

< Retour

[Déconnexion](#)

Rechercher

OK

Absence de Axel AIRE

Du / / à h

au / / à h

Commentaires :

Valider

Traiter les absences des élèves

Shrimp - Traiter les absences

[< Retour](#)
[Déconnexion](#)

Absences à traiter

Date de l'absence	Nom et prénom de l'élève	Détails
21/09/2015	SPARROW Jack	?
18/10/2015	MCQUEEN John	?
09/01/2016	STORM Susan	?

Shrimp - Traiter les absences

[< Retour](#)
[Déconnexion](#)

Absences à traiter

Absence de STORM Susan

Commentaires

5. Interfaces matérielles

L'application étant une application web, celle-ci n'a pas besoin de s'interfacer directement avec le matériel de la machine hôte. Elle utilisera notamment un serveur web (type Apache) afin d'accéder au réseau Internet, et un logiciel de gestion de base de données afin de communiquer avec le disque dur.

6. Interfaces logicielles

L'application s'exécutera sur une JVM 8 sur le système d'exploitation CentOS 7, dans sa version minimale. Elle sera connectée avec la base de données MySQL Community Server 5 par l'intermédiaire d'une classe Java. L'application utilisera le serveur web WildFly 8 développé par RedHat.

CAS :

Afin d'authentifier les utilisateurs de l'UL accédant à l'application, cette dernière s'interfacera en SSO avec le CAS. Pour ce faire, elle renverra au navigateur web de l'utilisateur le code de redirection 302 renvoyant vers l'url du serveur CAS de l'UL. Cette url contiendra en paramètre le service ID correspondant à l'application de gestion des absence, objet de ce projet.

ADE :

Les interfaces (accessibles par url) suivantes devront être utilisées par l'application :

(Toutes ces méthodes communiquent leurs résultats par trames XML)

- connect : Connexion à ADE. Renvoie le sessionId correspondant.

Paramètres obligatoires : login, password

- disconnect : Déconnexion de ADE.

Paramètres obligatoires : sessionId

- setProject : Sélectionne le projet courant

Paramètres obligatoires : sessionId, projectId

- getProjects : fournit une vue de l'ensemble des projets

Paramètres obligatoires : sessionId

- getResources : Retourne une liste ou un arbre de ressources du projet défini avec setProject (salles, etc)

Paramètres obligatoires : sessionId

- getActivities : Retourne une liste ou un arbre d'activités du projet défini avec setProject

Paramètres obligatoires : sessionId

- getEvents : Retourne une liste des événements d'événements survenus pendant la session

Paramètres obligatoires : sessionId

- imageEt : Cette méthode renvoie une image du planning au format gif.

Paramètres obligatoires : sessionId, resources, weeks, days

- getDate :

Paramètres obligatoires : sessionId, week, day, slot

Par ailleurs, seul l'utilisateur authentifié Administration aura la possibilité d'utiliser les homologues de ces méthodes permettant de modifier les données présentes dans la base de données de ADE.

7. Interfaces de communication

L'application web utilisera des protocoles web sécurisés pour assurer l'intégrité du système (les absences ayant un poids important sur l'obtention des diplômes) et sa confidentialité (les justificatifs étant privés). Le protocole HTTPS sera donc utilisé pour l'envoi de pages web et la récupération des pièces justificatives des étudiants depuis leur navigateur web jusqu'au serveur.

IV. Autres besoins non fonctionnels

1. Besoins liés aux performances

- L'application devra effectuer les requêtes dans la base de données et les afficher dans un délai acceptable (moins de deux secondes dans 99% des cas) afin d'éviter de perdre trop de temps au cours de l'utilisation, en particulier lorsque la scolarité choisi de valider toutes les absences d'une journée.
- Un appel complet (de la connexion à la validation de l'appel) pourra être effectué en moins de 3 minutes afin d'empiéter le moins possible sur le cours.
- Le serveur devra disposer d'au moins 8Go de mémoire vive. De plus, considérant que l'on a en moyenne 50% de toute la promotion (d'environ 33000 personnes) de l'Université de Lorraine qui est absente 5% du temps et qu'il faut conserver tous les justificatifs au moins 5 ans, en comptant 200 jours par année et 2 Mo par justificatif, il est nécessaire que le serveur dispose d'un disque dur d'au moins 10 To.
- L'application pourra mettre en cache dans le navigateur des images issues du site de Shrimp afin d'accélérer le chargement des pages lors des utilisations futures. Cependant pour des raisons de sécurité, on ne conservera aucun fichier d'absence ou de noms d'élève.
- L'application ne devra en aucun cas impacter sur les performances des autres applications en fonctionnement à l'UL.

2. Besoins liés à la sécurité

- L'application ne donne pas accès aux mêmes fonctionnalités aux utilisateurs selon leur groupe (professeur/scolarité/élèves/maintenance) pour des raisons de sécurité et de protection des données.
- L'application utilise une connexion sécurisée pour faire ses requêtes à la base de données afin de crypter les données envoyées.
- L'application gère de manière automatique la suppression des absences des étudiants datant de plus de 5 ans et qui en sont plus à l'UL.

3. Besoins liés à la qualité du produit

- L'application devra être compatible sur les principaux navigateurs internet : Internet 7 et supérieur, sa version Windows 10 Edge, Mozilla Firefox 40 et supérieur, Google Chrome, Safari et Opera. Par conséquent, l'application pourra être utilisée sur les systèmes d'exploitation Windows 7 et supérieur, MAC OS X et Linux.
- L'application mobile devra être compatible sur la plupart des smartphones possédant au moins un processeur Dual Core de 1.2GHz. De même elle sera disponible sur IOS, Android et Windows Phone.
- L'application devra pouvoir être exécutée sur des ordinateurs peu puissants afin que n'importe quel utilisateur puisse l'utiliser. Par ordinateur peu puissant, on sous-entend des ordinateurs ayant environ 2Go de mémoire vive.
- L'application devra pouvoir recevoir des mises à jour en cas de besoin tout en subissant un temps d'arrêt de moins d'une heure afin de perturber au minimum le fonctionnement de l'UL.

4. Autres besoins

- Pour permettre le fonctionnement de l'application, nous aurons besoin d'une base de données. Si possible nous utiliserons la base de données déjà présente dans le système informatique de l'université afin de faciliter son utilisation par le service informatique.
- L'application respectera le secret professionnel et l'anonymat, en aucun cas les absences ne seront diffusées au public.

Appendice A

Dictionnaire de données

Les absences pourront avoir, dans la base de données 5 états distincts.

- **A justifier** : l'absence est datée de moins de 15 jours et peut être encore justifiée
- **Non justifiée** : l'absence n'est plus justifiable
- **En attente** : l'élève a soumis une justification et la scolarité est en train de traiter cette dernière
- **Non valide** : la justification de l'absence a été refusée par la scolarité et l'élève peut fournir une nouvelle justification
- **Justifiée** : l'absence a été justifiée par l'élève et la scolarité a validé cette justification

Appendice B

Gestion de projet

L'élicitation des besoins fut l'objet de notre première réunion. Nous avons alors identifié les différents acteurs ainsi que leurs besoins fonctionnels propres (et les besoins non-fonctionnels associés). Chaque membre de l'équipe a élaboré sa vision du problème et l'a confrontée aux autres dans le but de créer une version commune.

Lors de la deuxième réunion, nous avons établi ensemble les différents scénarios que le système pourrait rencontrer face à ses utilisateurs. Nous les avons ensuite traduits sous forme de diagrammes d'activités. Une fois les exigences définies envers le système, nous avons décidé de l'architecture de notre solution.

Nous avons passé les réunions suivantes à rédiger les spécifications du système ébauché précédemment, chacun s'occupant d'un paragraphe différent et discutant des points critiques du système avec les autres membres de l'équipe.