

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE SOUSSE

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة



INSTITUT SUPERIEUR D'INFORMATIQUE
ET DES TECHNIQUES DE COMMUNICATION – HAMMAM SOUSSE

Département Multimédia

MEMOIRE DE STAGE DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de Mastère Professionnel en Service
Web et Multimédia

*Développement d'une Cross-
Plateforme d'enseignement en
ligne*

Réalisé par :
Hend Azzouna

Encadré par :
Mr. Yaroub ELLOUMI

Société d'accueil

Dev-GX

Année Universitaire 2020 – 2021

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE SOUSSE**

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة



**INSTITUT SUPERIEUR D'INFORMATIQUE
ET DES TECHNIQUES DE COMMUNICATION – HAMMAM SOUSSE**

Département Multimédia

MEMOIRE DE STAGE DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de Mastère Professionnel en Service
Web et Multimédia

Réalisé par :
Hend Azzouna

Encadrant : Mr. ELLOMI Yaroub

Signature :

Superviseur : Mr. Ben Ali Wassim

Signature :

Société d'accueil

Dev-GX

Année Universitaire 2020 – 2021

Dédicaces

Avec beaucoup d'amour je dédie ce travail

A mes chers parents,

Je ne trouve pas de mots pour exprimer mon respect et mon amour éternel pour eux, qui m'ont soutenue tout au long de ma carrière universitaire.

À ma chère mère

Est le meilleur exemple de ma vie et le secret de ma force. Dieu vous protège et vous donne la santé et une longue vie.

À mon cher frère

Pour l'amour et le Soutien,

Pour toutes les personnes qui m'ont donné leur amour et leur soutien moral et émotionnel.

✍️ Hend Azzouna

Remerciements

À la fin de ce projet, je suis heureux d'exprimer mes remerciements à tous ceux qui m'ont encouragé à mener à bien ce projet avec leur présence, leur soutien et leurs conseils.

Tout d'abord, je voudrais remercier Mr. Wassim Ben Ali, qui m'a donné l'expérience au sein de votre société DEV-GX pour m'avoir encadré et orienté durant mon stage PFE et l'occasion de développer mes connaissances.

Sincères mes remerciements à Mr. Yaroub ELLOMI mon encadrant au sein de l'institut supérieur d'informatique et des techniques de communication de Hammam Sousse (ISITCOM), qui m'aider toujours et soutenue dans mon travail.

Je tiens également à remercier toutes les personnes qui ont aidé de près ou de loin, contribué au succès de ce projet.

Enfin, je voudrais remercier les membres du jury qui m'ont honoré en accepter d'évaluer mon travail.

 Hend Azzouna

Table des matières

Introduction générale	1
Chapitre1 : Cadre général du projet	3
I. Introduction :.....	3
II. Présentation de l'entreprise :.....	3
III. Présentation du cadre de projet :.....	4
1. Etude de l'existant :	4
2. Critique de l'existant :.....	5
3. Solution proposée :	6
IV. Méthodologie de conception :.....	7
1. Méthodologie de gestion de projet agile « Scrum » :	7
2. Méthode de modélisation et de conception :	9
V. Conclusion :	9
Chapitre 2 : Initiation du projet (Sprint 0).....	10
I. Introduction :.....	10
II. Cadrage de besoins :.....	10
1. Analyse des besoins :	10
1.1. Besoins Fonctionnels :	10
1.2. Besoins non Fonctionnels :	11
2. Présentation des acteurs :	12
3. Diagramme de cas d'utilisation globale :.....	12
III. Le backlog Product :	13
IV. Architecture globale de l'application.....	15
V. Environnements de développement :.....	16
1. Langages et Framework Utilisés :	16
2. Environnement logiciel :	18
3. Environnement Mobile :	19
VI. Diagramme de classes de l'application	19
VII. Conclusion :	21
Chapitre 3 : Gestion d'accès d'utilisateurs et des publications (Sprint 1).....	22
I. Introduction.....	22
II. Le backlog du sprint 1 :.....	22
III. Cadrage de besoins :.....	24

1. Diagrammes des cas d'utilisation	24
1.1. Diagramme de cas d'utilisation « gérer équipe »	24
1.2. Diagramme de cas d'utilisation « gérer membre »	24
1.3. Diagramme de cas d'utilisation « gérer publication »	25
2. Description des cas d'utilisation	25
2.1. Description du cas d'utilisation S'authentifier	25
2.2. Description du cas d'utilisation Gérer les équipes	26
2.3. Description du cas d'utilisation Gérer les membres	27
IV. Conception	28
V. Réalisation	29
VI. Conclusion :	33
Chapitre 4 : Gestion des cours offline (Sprint 2)	34
I. Introduction	34
II. Le backlog du sprint 2 :	34
III. Cadrage de besoins :	35
1. Diagrammes des cas d'utilisation	35
1.1. Diagramme de cas d'utilisation « gérer cours »	35
1.2. Diagramme de cas d'utilisation « gérer chapitre »	36
1.3. Diagramme de cas d'utilisation « gérer évaluation »	37
2. Description des cas d'utilisation	37
2.1. Description du cas d'utilisation « gestion de cours »	37
2.2. Description du cas d'utilisation Gérer les chapitres	38
2.3. Description du cas d'utilisation Gérer les évaluations	39
IV. Conception	40
V. Réalisation	43
VI. Conclusion	50
Chapitre 5 : Gestion des cours online (sprint 3)	51
I. Introduction	51
II. Le backlog du sprint 3 :	51
III. Cadrage de besoins :	52
1. Diagrammes des cas d'utilisation	52
1.1. Diagramme de cas d'utilisation « gérer calendrier et visio-conférence »	52
1.2. Diagramme de cas d'utilisation « gérer messagerie instantanée »	53
2. Description des cas d'utilisation	54
2.1. Description du cas d'utilisation « gestion de visio-conférence »	54

2.2. Description du cas d'utilisation Gérer messagerie	56
IV. Conception	56
V. Réalisation	58
VI. Conclusion	63
Conclusion générale.....	64
Bibliographie	66

Table des figures

Figure 1 : logo de société.....	3
Figure 2: Organigramme de la société dev-gx.....	4
Figure 3: Architecture générale de projet global	7
Figure 4: le cycle itératif incrémentale	7
Figure 5 : Principe de la méthodologie Scrum.....	9
Figure 6: Diagramme de cas d'utilisation global	13
Figure 7: Planification des sprints	15
Figure 8: Modèle mvc.....	16
Figure 9: logo angular.....	16
Figure 10: logo nodejs	17
Figure 11: logo expressjs	17
Figure 12: logo mongodb.....	17
Figure 13: logo html	17
Figure 14: logo css	18
Figure 15: logo de javascript.....	18
Figure 16 : logo de Visual Studio code	18
Figure 17: Architecture logicielle de l'application.....	19
Figure 18: Diagramme de cas d'utilisation de "gérer équipe"	24
Figure 19: diagramme de cas d'utilisation de "gérer membre"	24
Figure 20: Diagramme de cas d'utilisation de "gérer publication"	25
Figure 21: Diagramme de séquence s'authentifier.....	28
Figure 22: Diagramme de séquence Ajouter équipe	29
Figure 23: Interface Login	30
Figure 24: interface "accueil".....	30
Figure 25: Interface "Gestion des équipes"	31
Figure 26: Interface "Ajouter équipe"	31
Figure 27: Interface "Rejoindre équipe"	32
Figure 28: Interface "Gestion des membres"	32
Figure 29: Interface "gestion des publications et des commentaires"	33
Figure 30: Diagramme de cas d'utilisation "Gestion des cours"	36

Figure 31:Diagramme de cas d'utilisation "Gestion des chapitres"	36
Figure 32:Diagramme de cas d'utilisation "Gestion des évaluations"	37
Figure 33:Diagramme de séquence Créer cours	41
Figure 34:Diagramme de séquence Publier quiz	42
Figure 35:Diagramme de séquence Supprimer chapitre.....	43
Figure 36:Interface "Gestion des cours"	43
Figure 37:Interface " Ajouter cours"	44
Figure 38:Interface "Contenu cours pour responsable"	44
Figure 39:Interface " Contenu cours pour participant"	45
Figure 40:Interface "Gestion de chapitre"	45
Figure 41:Interface "charger fichier"	46
Figure 42 : interface gestion de quiz	46
Figure 43:Interface "Gestion des quiz"	47
Figure 44:Interface "Ajouter question"	47
Figure 45:Interface "Publier Quiz"	48
Figure 46:Interface "Modifier question"	48
Figure 47:Interface " Passer Test"	49
Figure 48:Interface "Test"	49
Figure 49:Interface " Réponse correct"	50
Figure 50:Interface " Test"	50
Figure 51:Diagramme de cas d'utilisation "Gestion de calendrier et visio-conférence"	53
Figure 52:Diagramme de cas d'utilisation de gestion de messagerie	54
Figure 53:Diagramme de séquence "Envoyer un message"	57
Figure 54:Diagramme de séquence " Gestion de notification"	58
Figure 55:Interface "Gestion de messagerie"	58
Figure 56:Interface "Gestion de notification"	59
Figure 57:Interface " Gestion de calendrier"	59
Figure 58:Interface "Planifier visio-conférence	60
Figure 59:Interface " Modifier visio-conférence"	60
Figure 60:Interface " Supprimer visio-conférence"	61
Figure 61:Interface "démarrer une visio-conférence "	61
Figure 62:Interface "accepter ou refuser la participation"	62
Figure 63:interface de participant	62
Figure 64:interface de l'instructeur.....	63

Liste des tableaux

Tableau 1:fiche technique	3
Tableau 2: tableau des applications existantes.....	5
Tableau 3: comparaison des plateforme d'enseignement en ligne	5
Tableau 4: Backlog product.....	14
Tableau 5 : backlog du sprint1	23
Tableau 6: Description de cas d'utilisation Authentification	26
Tableau 7: description de cas d'utilisation crée équipe	27
Tableau 8: Description de cas d'utilisation ajouter membre	27
Tableau 9 : backlog du sprint 2.....	35
Tableau 10:Description de cas d'utilisation créer un cours	38
Tableau 11:Description de cas d'utilisation publier une ressource	39
Tableau 12:Description de cas d'utilisation publier quiz.....	40
Tableau 13: Backlog du sprint3	52
Tableau 14:Description de cas d'utilisation planifier visio-conférence.....	55
Tableau 15 : description de cas d'utilisation modifier visio-conférence	55
Tableau 16:Description de cas d'utilisation envoyer message	56

Liste des abréviations

TIC : Technologies d'Information et de Communication

CSS : Cascading Style Sheets

HTML : HyperText Markup Language

JS : JavaScript.

MVC : Modèle Vue Contrôleur

UML : Unified Modeling Language

XP : eXtreme Programming

RUP : Rational Unified Process

NOSQL : Not Only SQL

Introduction générale

L'enseignement est un mode d'éducation permettant de développer les connaissances d'un élève par le biais de la communication verbale et écrite. De nos jours, vu que les moyens de communication sont évolués, alors il existe deux modes d'enseignement qui sont le mode traditionnel et le mode moderne.

L'enseignement traditionnel est un enseignement classique, centré sur le cours présentiel et aussi sur le cours magistral. Il résulte la présence de l'instructeur et l'apprenant au sein d'un seul et même lieu selon un emploi de temps précis. Un cours magistral associé à des travaux dirigés et pratiques et les supports sous forme de recueil de textes, de livres et autres documents imprimés et des explications au tableau.

L'enseignement moderne est un enseignement en ligne qui se déroule sur internet. Il permet de fournir un cours en temps et lieu synchrone et asynchrone et faciliter l'accès à distance à une salle de classe. Les apprenants n'ont plus besoin de se déplacer à l'école, il exige des moyens pour faciliter l'apprentissage de l'apprenant. Chaque apprenant s'adapte à son propre rythme de travail.

Il existe plusieurs solutions d'éducation en ligne. Ces solutions utilisent les outils de TIC afin d'assurer le processus d'apprentissage en ligne. Il existe deux modes d'enseignement en ligne qui sont l'enseignement en mode offline qui permet le partage des informations sous forme de PDF, vidéo ... et l'enseignement en mode online qui permet la visio-conférence, chat.... Cependant, elles possèdent plusieurs inconvénients comme il y a deux modes d'enseignement, le nombre limité des apprenants et le temps limité pour la visio-conférence.

C'est dans ce cadre que se situe mon projet de fin d'études intitulé « Développement d'une application web et mobile pour l'enseignement en ligne », réalisé au sein de l'entreprise « DEV-GX » et à l'institut supérieur d'informatique et des techniques de communication de Hammam Sousse. Le projet consiste à créer une application web et mobile qui permet l'enseignement on-line et l'enseignement off-line avec les nouvelles technologies de l'information qui offrent un large champ de possibilités.

Pour faciliter le travail, j'ai choisi la méthodologie agile scrum qui permet le travail en équipe, chacun son rôle dans l'équipe pour structurer le travail.

Plusieurs utilisateurs qui peuvent accéder au même temps à notre application pour cela j'ai choisi l'environnement Angular pour le développement de la partie front-end, Nodejs pour le développement de la partie backend et MongoDB pour la base de données.

Ainsi, nous articulons notre rapport autour de cinq chapitres.

Le premier chapitre mettra notre projet dans son contexte, nous y trouverons une présentation de l'organisme qui nous a accueilli, la problématique, et en fin le travail demandé et la méthodologie qui nous a permis de la réaliser

Le deuxième chapitre aborde le fameux sprint 0, qui constitue la phase d'initiation d'un projet Scrum. Il traite d'abord le recueil des besoins, et l'élaboration du Backlog du produit, ensuite, la mise en place de l'architecture de notre application, et la préparation de l'environnement de développement.

Les trois derniers chapitres, auront le même plan : un Backlog Sprint, cadrage des besoins, la conception et la réalisation, qui reflètent le déroulement des itérations. Le premier sprint s'occupe de la partie gestion d'accès d'utilisateur des publications. Le deuxième abordera la gestion des cours off-line qui contient la gestion des cours, la gestion des chapitres, la gestion des ressources et la gestion d'évaluation. Enfin, le troisième sprint mettra en lumière la partie gestion des cours on-line qui contient la gestion du calendrier et visio-conférence, la gestion des messageries et la gestion des notifications.

Nous finirons par une conclusion générale qui récapitulera le travail élaboré et qui essaiera de révéler certaines perspectives pour son amélioration.

Chapitre1 : Cadre général du projet

I. Introduction :

Dans ce chapitre nous allons présenter l'entreprise Dev-GX, consulter ses meilleurs services et analyser son environnement interne et externe. Ensuite, nous décrirons brièvement le sujet et le travail demandé par Dev-GX. Enfin nous présenterons la méthodologie de travail adoptée.

II. Présentation de l'entreprise :

Dev GX est basée dans l'une des principales villes de la Tunisie, Sousse. L'équipe met tous ses efforts et son expertise pour fournir des services sans faille à ses clients. Ce qui facilite la construction de leurs projets. Dev-GX est une société spécialisée depuis 2010, dans le développement, la commercialisation, la mise en place et la formation de progiciels de gestion pour les entreprises. Ces logiciels sont élaborés avec des critères très rigoureux citant :

- La simplicité qui permet une manipulation facile.
- L'efficacité qui permet la satisfaction des besoins de client.
- La sécurité qui permet la protection des données.

Ces logiciels peuvent être :

- Des applications Web.
- Des applications Mobiles (Android, IOS, Windows Phone).
- Des applications bureautiques (Desktop).



FIGURE 1 : LOGO DE SOCIETE

La fiche technique est représentée dans le tableau suivant :

➤ Nom de l'organisme :	➤ Dev-GX
➤ Adresse :	➤ Sousse Bab Bhar
➤ Email :	➤ Dev-GX@gmail.com
➤ Gérant :	➤ Wassim Ben Ali
➤ Numéro téléphone portable :	➤ 31 570 517

TABLEAU 1:FICHE TECHNIQUE

L'organisation de la société Dev-GX est représentée dans l'organigramme suivant :

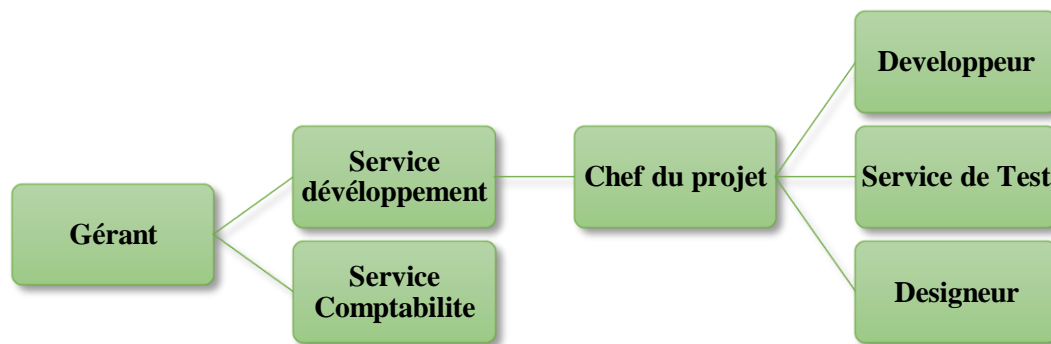


FIGURE 2: ORGANIGRAMME DE LA SOCITET DEV-GX

III. Présentation du cadre de projet :

Durant les dernières années, le monde a connu plusieurs changements, évolutions, et phénomènes des nouvelles technologies, comme l'utilisation des Technologies d'Information et de Communication « TIC ». L'enseignement en ligne consiste à mettre en pratique les outils de TIC pour assurer le processus d'apprentissage en ligne. Il permet aux apprenants de suivre des cours, des formations continues ou des formations spécifiques. Grâce à l'ère numérique, il existe plusieurs applications pour l'enseignement en ligne qui assure une présence virtuelle de l'étudiant et l'instructeur ... Suite au développement de ces applications, nous présenterons notre application qui consiste à développer une application web et mobile pour l'enseignement en ligne. Pour cela dans cette partie-là, nous commencerons par présenter l'existant. Ensuite, nous détaillerons ses limites. Enfin, nous montrerons notre solution.

1. Etude de l'existant :

L'étude de l'existant est une phase importante. Alors, après avoir émis le besoin de gérer l'enseignement en ligne via une application, il convient de procéder à une étude de l'existant, ne serait-ce que pour savoir s'il vaut mieux recourir à une solution externe ou alors développer son propre outil. **Zoom**, **Google Meet** et **Google Classroom** sont actuellement parmi les outils les plus connus pour l'enseignement en ligne.




	Zoom	Google meet	Google classroom
Description	Zoom est un logiciel de visioconférence à distance qui permet la réalisation de conférences vidéo par le biais d'une interface ergonomique et structurée. Il correspond à l'enseignement online.	Google meet permet de créer des réunions vidéo et accéder à chaque salle de conférence, il est très simple en pratique. Il correspond à l'enseignement online.	Un outil de collaboration conçu pour les enseignants et leurs élèves qui permet aux éducateurs de créer des devoirs, de recevoir des commentaires et de suivre d'autres informations à partir d'un emplacement centralisé. Il correspond à l'enseignement offline.
Logo			

TABLEAU 2: TABLEAU DES APPLICATIONS EXISTANTES

2. Critique de l'existant :

L'étude de l'existant nous a permis de dégager un certain nombre de lacunes :

Fonctionnalité	Zoom	Google meet	Google classroom
Apprentissage synchrone et asynchrone	Synchrone	Synchrone	Asynchrone
Temps limité	40 minutes	60 minutes	
Nombre maximale de participant	500 personnes	250 personnes	
Propose une version web et mobile	✓	✓	✓
Partage de fichier	✓	✓	✓
Chat individuel, avec groupe	✓		
Création et partage de devoir			✓
Partage webcam	✓	✓	
Partage d'écran	✓	✓	

TABLEAU 3: COMPARAISON DES PLATEFORME D'ENSEIGNEMENT EN LIGNE

Comme le montre le tableau ci-dessus, les solutions existantes d'enseignement en ligne sur le marché proposent différentes fonctionnalités de base. Cependant, elles possèdent des inconvénients comme : Il existe deux modes d'enseignement : enseignement en mode online (synchrone) et enseignement en mode offline (asynchrone).

- L'enseignement online permet le partage d'écran, webcam, chat. Il offre un nombre limité de participant et un temps limité de vidéo conférence pour tous les systèmes existants. Il est caractérisé par un coût élevé vue la nécessité de la disponibilité de l'enseignant, même si les contenus sont redondants.
- L'enseignement offline permet le partage des fichiers ou la publication d'une information et la création des devoirs. Il est caractérisé par des contenus statiques, parfois non adaptés aux besoins.

Notre étude a montré que les solutions du marché n'offrent pas les fonctionnalités nécessaires à une plateforme d'enseignement en ligne et non pas très professionnel.

3. Solution proposée :

Dans ce cadre Dev GX souhaite développer une application web et mobile qui assure l'enseignement en ligne pour tous types d'organismes. Notre Cross-Plateforme doit assurer l'enseignement en ligne. L'originalité de l'application consiste à combiner les fonctionnalités des deux modes d'enseignement :

- ✓ Le premier mode est contenu des contenus standard qui est l'enseignement en mode off-line qui contient la gestion des chapitres et des évaluations.
- ✓ Le deuxième mode est contenu des formations spécifiques aux participants qui est l'enseignement en mode on-line qui contient la gestion de visio-conférence, discussion instantanée et la gestion de calendrier pour les planifications.

Notre application consiste aussi la gestion d'accès d'utilisateurs et des publications qui contient l'authentification, la gestion des membres, des équipes, des publications et des commentaires, on a aussi la gestion des notifications. Cette Cross-Plateforme destiné pour toutes les facultés, les écoles et les collèges soit privé soit étatique, aussi pour les écoles de formations et des clubs. Pour cela, notre application sera développée pour les utilisations du web et du mobile.

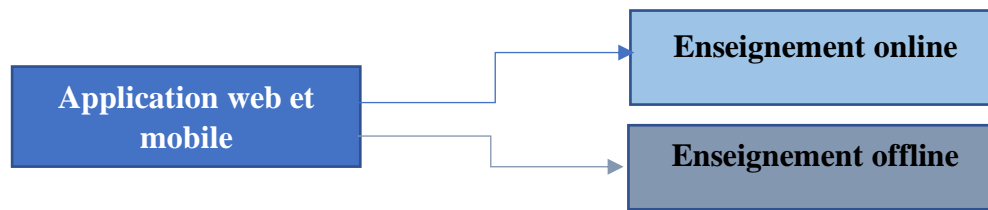


FIGURE 3: ARCHITECTURE GENERAL DE PROJET GLOBAL

IV. Méthodologie de conception :

Le plus grand avantage d'une méthode orientée objet est qu'elle permet de structurer un système sans centrer l'analyse uniquement sur les données ou uniquement sur les traitements mais sur les deux à la fois. Une telle approche a pour but de modéliser les propriétés statiques et dynamiques de l'environnement du système. Elle met en correspondance le problème et la solution, en préservant la structure et le comportement du système analysé. Ceci, nous a conduit à adopter l'approche orientée objet pour modéliser notre système en se basant sur les diagrammes Scrum.

1. Méthodologie de gestion de projet agile « Scrum » :

Avec Agile on ne parle pas de cycle :

- ✓ C'est surtout une question de philosophie.
- ✓ De mentalité et de culture qui place la réactivité, la collaboration et l'implication du client en priorité.
- ✓ Les méthodes Agile ont une approche itérative et incrémentale.

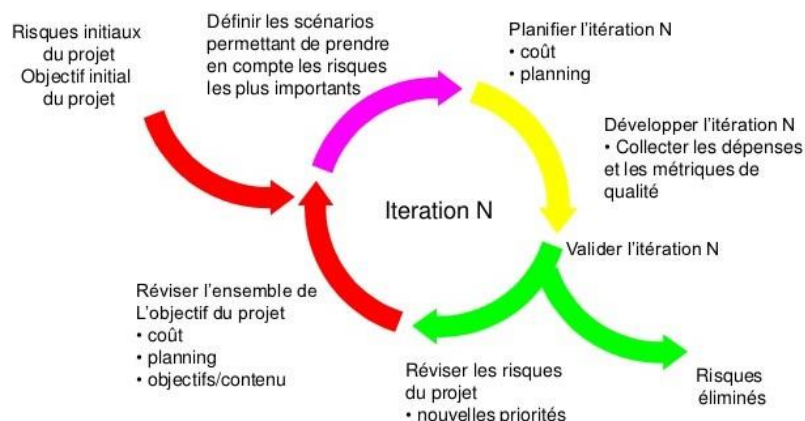


FIGURE 4: LE CYCLE ITERATIF INCREMENTALE

Il y a plusieurs méthodologies Agiles, les plus utilisées dans le marché sont :

- ✓ Kanban
- ✓ XP
- ✓ Scrum
- ✓ RUP

Afin d'accomplir notre application, nous avons utilisé la **méthode agile Scrum** pour l'appliquer au bout de la réalisation de ce projet. Scrum met en exergue l'esprit d'équipe et la volonté d'avancer ensemble vers un même but. Scrum est une méthode itérative incrémentale centrée sur la réalisation d'un produit fonctionnel.

Avantages de la méthode Scrum :

- ✓ Fortement inspirées de celles du Lean management
- ✓ Eliminer les gaspillages
- ✓ Favoriser l'apprentissage et les changements
- ✓ Reporter la prise de décision
- ✓ Livrer rapidement le produit
- ✓ Responsabiliser les membres d'équipe
- ✓ Construire la qualité tout au long du projet

Il existe trois acteurs clés qui sont :

- Le Product Owner : il a une vision complète du produit
- Le Scrum master : il assure le bon déroulement du projet et rappelle la méthode suivie
- L'équipe Scrum : elle développe le produit

La figure ci-dessous représente le principe de fonctionnement de la méthodologie Scrum :

- ✓ Premièrement, le dégagement du maximum des fonctionnalités à réaliser pour former le Back log du produit.
- ✓ Deuxièmement, la définition des priorités des fonctionnalités et de trier lesquelles seront réalisées dans chaque lancement.
- ✓ Finalement, la focalisation de groupe sur toutes les fonctionnalités à réaliser, dans des itérations appelées sprints.

Dans chaque sprint, un produit fonctionnel est livré et est appelé incrément.

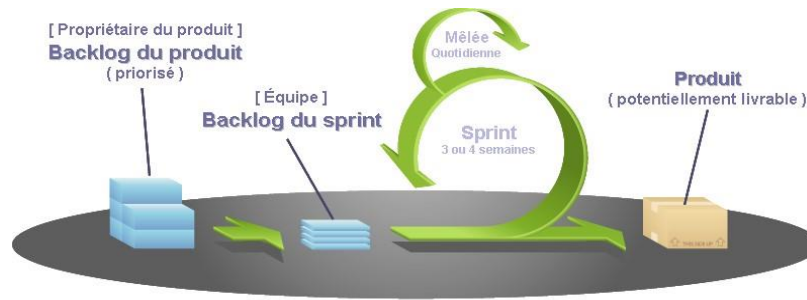


FIGURE 5 : PRINCIPE DE LA METHODOLOGIE SCRUM

2. Méthode de modélisation et de conception :

La conception de notre projet est basée sur le concept objet à l'aide d'UML (Unified Modeling Language) qui est par principe un langage de modélisation standardisé pour la conception des systèmes informatiques. Nous avons choisi l'UML car il n'impose pas une méthode de travail particulière. Il présente un ensemble d'outils pour améliorer les méthodes de travail tout en gardant les modes de fonctionnement. UML est un langage des modélisations très intuitif, simple et cohérent.

Il représente des différents points de modélisation qui permettent la visualisation et la manipulation des éléments. Aussi, il permet la limitation de plusieurs erreurs. Nous avons utilisé l'outil StarUML pour schématiser des systèmes complexes sous un format graphique et textuel simple et normal.

UML comporte trois grands groupes qui sont :

- Diagrammes structurels ou diagrammes statiques (UML Structure)
 - ✓ Diagramme de classes (Class diagram)
- Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)
 - ✓ Diagramme de cas d'utilisation (Use case diagram)
- Diagrammes d'interaction (Interaction diagram)
 - ✓ Diagramme de séquence (Sequence diagram)

V. Conclusion :

Après avoir mis le projet dans son cadre et après avoir mis en place une démarche de développement qui nous aidera tout au long du projet, nous pouvons ainsi entamer la prochaine étape c'est-à-dire décortiquer le cahier de charges pour analyser les besoins qui s'y trouvent et ainsi les spécifier.

Chapitre 2 : Initiation du projet (Sprint 0)

I. Introduction :

Vu que nous avons achevé la première phase du cycle de développement, nous aborderons dans ce chapitre la deuxième phase qui se concentre essentiellement sur la modélisation des besoins fonctionnels et non fonctionnels, présentation des acteurs, identifier les cas d'utilisation et réaliser le diagramme des cas d'utilisation général. Puis, nous produirons le backlog de l'application et le plan de réalisation. Ensuite, nous présenterons l'architecture générale de notre application, l'environnement matériel, les technologies et les langages de programmation utilisés pour la mise en place de l'environnement de travail. Enfin, nous aborderons la partie déploiement de l'application et son architecture physique.

II. Cadrage de besoins :

Au cours de cette étape, nous allons lister et détailler les besoins qui présenteront cette application, nous allons identifier par la suite les différents utilisateurs et nous finirons par présenter le diagramme des cas d'utilisation global de l'application.

1. Analyse des besoins :

1.1. Besoins Fonctionnels :

- **Gestion d'accès d'utilisateurs et des publications** : cette gestion est composée par :
 - ✓ **Authentification** : L'utilisateur ne peut accéder à l'application qu'après l'authentification.
 - ✓ **Gestion des équipes** : Après l'authentification, le responsable peut créer une équipe, consulter la liste de ses équipes, chercher une équipe par sa clé, demande de rejoindre l'équipe pour consulter l'équipe qu'il convient.
 - ✓ **Gestion des membres** : Gérer les membres par l'acceptation ou le refus d'une demande et l'ajout d'un nouveau membre à l'équipe ou mettre exclu un membre
 - ✓ **Gestion des publications** : les utilisateurs peuvent gérer les publications par l'ajout, la consultation et la suppression d'une publication.
 - ✓ **Gestion des commentaires** : les utilisateurs peuvent gérer les commentaires par l'ajout, la consultation et la suppression d'un commentaire.

- **Gestion des cours offline** : ce cours est représenté par :
 - ✓ **Gestion des cours** : Le responsable permet de créer un nouveau cours, d'affecter un instructeur au cours et de consulter la liste des cours.
 - ✓ **Gestion des chapitres** : L'instructeur peut ajouter, supprimer un chapitre et consulter la liste des chapitres pour chaque cours. Après la création d'un chapitre, l'instructeur peut ajouter une nouvelle ressource sous forme de mp4, PDF, DOC ou image pour chaque chapitre.
 - ✓ **Gestion d'évaluation** : contient la gestion des quiz et des réponses : L'instructeur peut créer un nouveau quiz pour chaque chapitre, publier, supprimer et modifier un quiz. Après la publication de quiz, le participant peut ajouter une réponse et l'instructeur peut consulter les réponses des participants.
- **Gestion des cours online** : ce cours est représenté par :
 - ✓ **Gestion du calendrier et visio-conférence** : L'instructeur peut planifier, modifier et supprimer des séances de cours online sur le calendrier. Ainsi, il peut démarrer une visio-conférence. Le participant peut accepter ou refuser la participation à cette visio-conférence.
 - ✓ **Gestion des messageries** : les utilisateurs peuvent discuter entre eux via une messagerie instantanée (chat/appe)l)
 - ✓ **Gestion des notifications** : Système de notification en temps réel pour la publication des quiz, l'affectation d'instructeur au cours et la planification de cours online.

1.2. Besoins non Fonctionnels :

Aussi important que les besoins fonctionnels, le service livré doit assurer les exigences non fonctionnelles tel que :

- **Sécurité** : L'application doit respecter la confidentialité des données. Du coup, les données personnelles des utilisateurs et les informations concernant les partenaires doivent être bien protégées.
- **Performance** : L'application doit être avant tout performante. C'est-à-dire à travers ses fonctionnalités, elle doit répondre à toutes les exigences des usagers d'une manière optimale.
- **Maintenabilité** : L'application doit être facile à maintenir pour l'utilisateur. C'est pour ça, le code doit être bien structuré et lisible.

- **Ergonomie** : L'application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.
- **Portabilité** : La solution doit être portable, pouvant s'exécuter sans configuration supplémentaire.

2. Présentation des acteurs :

Un acteur joue un rôle d'une entité extérieure qui interagit directement avec le système. Cette application présente principalement par **trois acteurs** :

Responsable d'établissement : C'est un utilisateur qui possède un compte et qui gère les membres (accepter, refuser les demandes des membres, ajouter et mettre exclu un membre), les équipes (créer une équipe) et les cours (Ajouter, affecter un instructeur a un cours).

Instructeur : C'est un utilisateur qui possède un compte et qui gère les cours offline (consulter le contenu de cours, gérer les chapitres (ajouter, modifier ou supprimer un chapitre), gérer les ressources, gérer les quiz (créer, publier, modifier un quiz), consulter les réponses des participants/quiz) et gère les cours online (gérer calendrier et visio-conférence (planifier, annuler, modifier, démarrer visio-conférence)).

Participant : C'est un utilisateur qui possède un compte et qui demander de rejoindre une équipe, consulter une équipe, visualiser les cours et les chapitres, répondre au quiz, participer à une visio-conférence, recevoir des notifications en temps réel et communiquer avec les autres utilisateurs via une messagerie instantanée (chat, appel).

3. Diagramme de cas d'utilisation globale :

Les diagrammes de cas d'utilisation expriment les fonctionnalités du système selon le point de vue de l'utilisateur et ils décrivent le système et les relations entre le système et son environnement. Alors, nous présentons le diagramme de cas d'utilisation global qui illustre les différentes fonctionnalités et les acteurs de notre application.

Les taches de gestion des publications, des commentaires et de messagerie instantanée nécessite une authentification de l'utilisateur. Pour cela elles sont tous liées par s'authentifier par une relation include. Ensuite, on a le responsable est un utilisateur qui doit être authentifier pour gérer les équipes, les membres et les cours. Pour cela les cas d'utilisations associé sont en relations include avec le cas s'authentifier.

L'instructeur est un utilisateur qui doit être authentifié pour gérer les cours-offlines, les cours online, le calendrier et les notifications. Pour cela elles sont tous liées par s'authentifier par une relation include. Enfin, on a le participant qui est un utilisateur qui doit être authentifié pour demande de rejoindre une équipe, de participer à un cours online et aussi pour répondre au quiz. Pour cela les cas d'utilisations associé sont en relations include avec le cas s'authentifier.

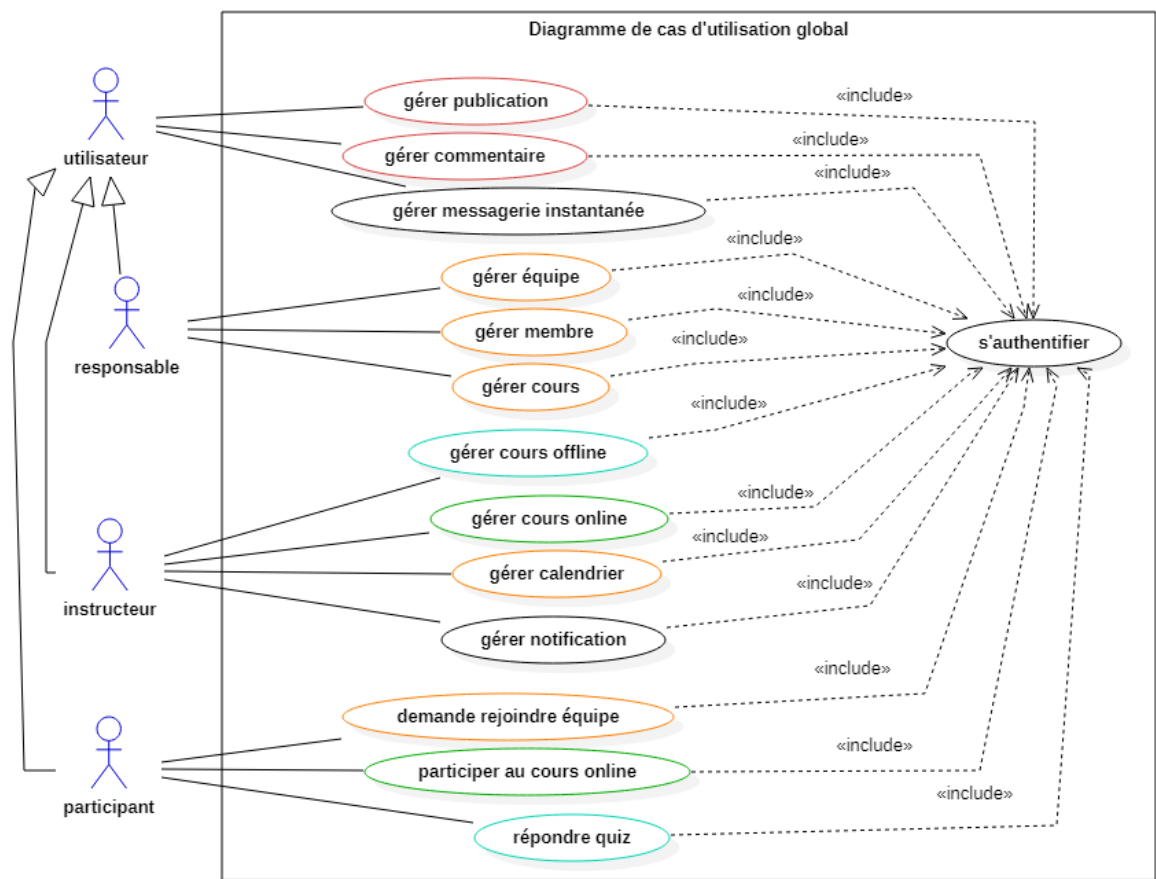


FIGURE 6: DIAGRAMME DE CAS D'UTILISATION GLOBAL

III. Le backlog Product :

Le Backlog est un artéfact très important dans Scrum. Il dresse toutes les fonctionnalités désirées par les futurs utilisateurs et exprimées sous forme d'User Stories

Le tableau suivant présente le back log de notre produit.

- ID qui représente l'identifiant de l'use story
- Thème pour mieux ordonner les user stories
- Scenario ou User Story comporte la description des user story

- La priorité de l'user story selon la valeur métier et l'ordre de réalisation.

Id sprint	Thème	ID	Acteur	User Story	Priorité
1	Gestion d'accès d'utilisateurs et des publications	1.1	Utilisateur	En tant qu'utilisateur, je suis obligé de s'inscrire puis de s'authentifier à l'application.	Elevée
		1.2	Utilisateur	En tant qu'utilisateur, je peux quitter l'application.	Elevée
		1.3	Responsable	En tant que responsable d'établissement, je veux gérer les membres.	Elevée
		1.4	Responsable	En tant que responsable d'établissement, je veux gérer les équipes.	Elevée
		1.5	Utilisateur	En tant qu'utilisateur, je veux gérer les publications.	Elevée
		1.6	Utilisateur	En tant qu'utilisateur, je veux gérer les commentaires.	Elevée
	Gestion des cours off-line	1.8	Instructeur	En tant qu'instructeur, je veux gérer les chapitres.	Elevée
		1.9	Instructeur	En tant qu'instructeur, je veux gérer les ressources.	Elevée
		1.10	Instructeur	En tant qu'instructeur, je veux gérer les évaluations.	Elevée
	Gestion des cours on-line	1.12	Instructeur	En tant qu'instructeur, je veux gérer le calendrier et la visio-conférence.	Elevée
		1.13	Utilisateur	En tant qu'utilisateur, je veux gérer les messageries instantanées.	Elevée
		1.14	Instructeur	En tant qu'instructeur, je veux gérer les notifications.	Elevée

TABLEAU 4: BACKLOG PRODUCT

Planification des sprints :

Lors de la terminaison de backlog de produit, nous passons à l'établissement de la réunion de planification dont le but est de construire le backlog de sprint qui se base sur le backlog de produit. Finalement, nous identifierons des durées prévisionnelles du travail à effectuer pendant chaque sprint. Notre travail sera divisé sur trois sprints :

- Le premier Sprint va traiter la partie « gestion d'accès d'utilisateurs et des publications ». Après l'authentification, l'utilisateur peut « gérer les équipes », « gérer les membres » et de « gérer les publications et les commentaires ».
- Le deuxième sprint s'intéressera à la « gestion des cours off-line » qui contient la « gestion des chapitres ». Le cours doit avoir au moins un chapitre, et la « gestion d'évaluation ».
- Le dernier sprint traitera la « gestion des cours on-line » qui contient « la gestion du calendrier et de visio-conférence », la « gestion des messagerie » et la « gestion des notifications ».

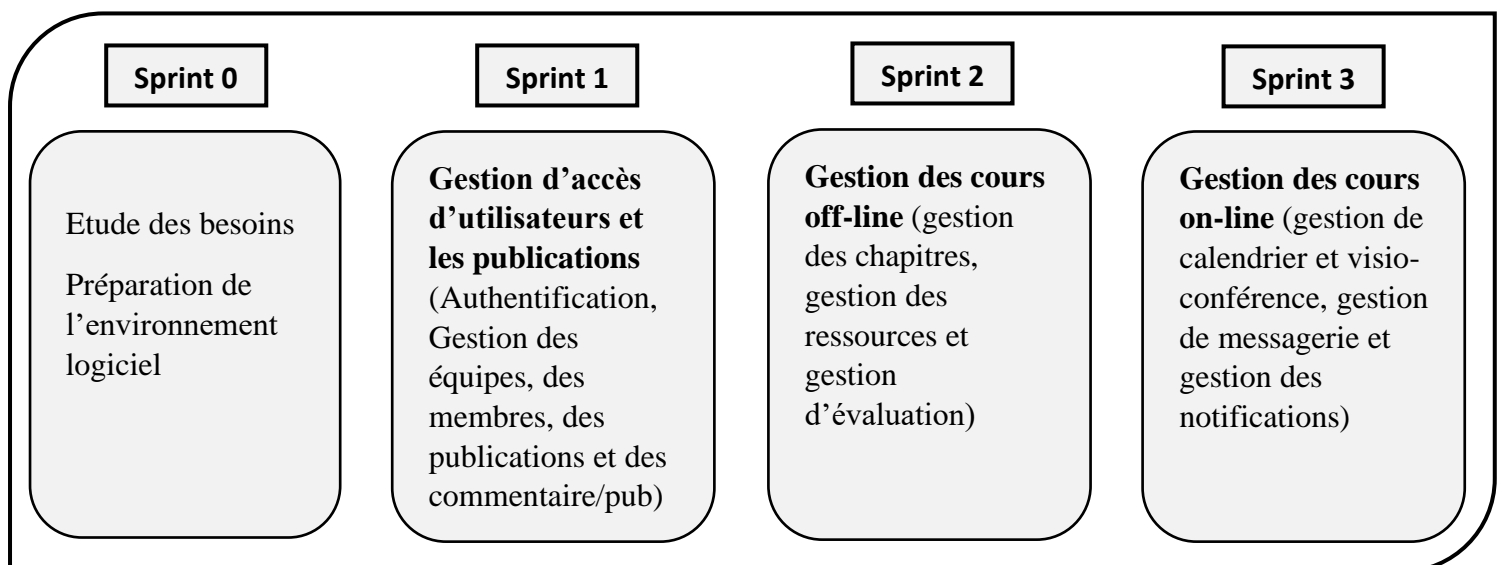


FIGURE 7: PLANIFICATION DES SPRINTS

IV. Architecture globale de l'application

L'architecture que nous avons utilisée est Angular qui implémente le **design pattern MVC** (Modèle-Vue-Contrôleur) qui est un motif d'architecture logicielle le plus utilisée pour les applications web. **MVC** permet de bien organiser le code source, de savoir quel rôle joue précisément chaque fichier dans l'application et consiste à s'éparer la logique du code en trois parties :

Modèle : la partie qui concerne les données (base de données, fichiers...) de l'application (site...). Il dispose de toutes les méthodes d'accès aux données (insert, delete, update, select)

Vue : la partie qui concerne l'affichage : l'interface avec laquelle l'utilisateur interagit (page HTML...). C'est la vue qui s'occupe de la visualisation des données retournées par le modèle

Contrôleur : c'est l'intermédiaire entre le modèle et la vue. Il demande les données au modèle, les analyse, et prend ensuite des décisions et renvoie le texte à afficher à la vue.

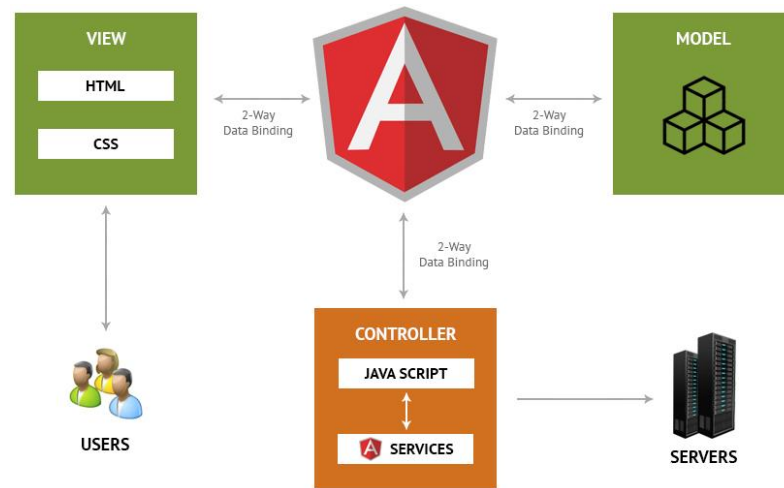


FIGURE 8:MODELE MVC

V. Environnements de développement :

Pour la réalisation de ce projet nous avons choisi de travailler avec :

1. Langages et Framework Utilisés :

- **Angular 11** : est un **Framework** open source écrit en **JavaScript** qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « *Single Page Applications* » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.



FIGURE 9: LOGO ANGULAR

NodeJS : est une plateforme construite sur le moteur JavaScript V8 de Chrome qui permet de développer des applications en utilisant du JavaScript. Il se distingue des autres plateformes

grâce à une approche non bloquante permettant d'effectuer des entrées/sorties (I/O) de manière asynchrone.



FIGURE 10: LOGO NODEJS

- **ExpressJS** : est un framework pour construire des applications web basées sur Node.js. C'est de fait le framework standard pour le développement de serveur en Node.js. ExpressJS est une librairie qui vous permettra de créer une application Web plus simplement qu'avec l'objet http directement. Elle fournit un ensemble de méthode permettant de traiter les requêtes HTTP et fournit un système de middleware pour étendre ses fonctionnalités.



FIGURE 11: LOGO EXPRESSJS

- **MongoDB** : est une base de données distribuée, universelle et basée sur des documents, qui a été conçue pour les développeurs d'applications modernes et pour l'ère du Cloud. Aucune autre base de données n'offre une telle productivité. MongoDB est base de données de documents, ce qui signifie qu'elle stocke les données au format de documents JSON. Nous estimons qu'il s'agit de la façon la plus naturelle d'envisager les données, bien plus efficace et expressive que le modèle traditionnel basé sur des rangées et des colonnes.



FIGURE 12: LOGO MONGODB

- **HTML 5** : est un langage de balisage qui permet de créer des pages Web. Les balises permettent de mettre en forme le texte et de mettre en place des composants interactifs, tels que des liens, des boutons, des formulaires, des courbes graphiques ainsi que des animations et des vidéos



FIGURE 13: LOGO HTML

- **CSS3** : est utilisé avec HTML pour créer et formater la structure du contenu. Il est responsable des couleurs, des propriétés des polices, des alignements de texte, des images d'arrière-plan, des graphiques, des tableaux, etc. Il fournit le positionnement de divers éléments avec des valeurs fixes, absolues et relatives.



FIGURE 14: LOGO CSS

- **JavaScript** : est un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser. Bon, peut-être pas tout, mais vous pouvez faire bien des choses avec quelques lignes de JavaScript.



FIGURE 15: LOGO DE JAVASCRIPT

2. Environnement logiciel :

Dans cette partie nous allons présenter l'environnement logiciel utilisé pour réaliser notre projet.

- **Visual Studio code** : est un éditeur de code multiplateforme édité par Microsoft. Cet outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le HTML, C++, PHP, Javascript, Markdown, CSS, etc



FIGURE 16 : LOGO DE VISUAL STUDIO CODE

- **Android Studio** : est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux

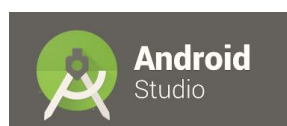


FIGURE 7 : LOGO DE ANDROID STUDIO

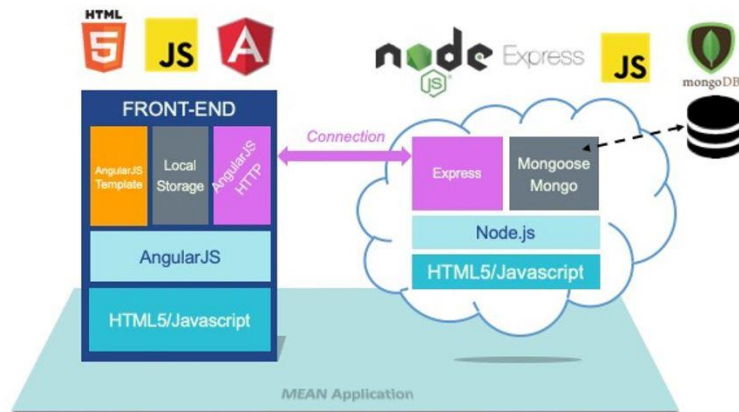


FIGURE 17: ARCHITECTURE LOGICIEL DE L'APPLICATION

3. Environnement Mobile :

Cordova est un Framework de développement mobile hybride open-source de la fondation Apache. Il permet d'exploiter les technologies Web courantes telles que HTML5, CSS3 et JavaScript pour développer des applications multi-plateformes et offrir un accès aux fonctionnalités du dispositif mobile via une API Javascript.



FIGURE 8 : LOGO DE APACHE CORDOVA

VI. Diagramme de classes de l'application

Cette partie illustre le diagramme de classe de la totalité du projet. Ces différentes classes représentent les éléments qui composent le modèle de données et leurs relations.

La présentation des utilisateurs suivant leurs rôles est élaborée par ces objets suivants : on a la classe utilisateur qui définit les utilisateurs qui interviennent à notre application. Les classes responsable, instructeur et participant qui sont héritées de la classe utilisateur.

Ensuite, la gestion de nos équipes est présentée par la classe équipe qui permet la composition d'une ou plusieurs formations et la planification du calendrier par ces objets. On a aussi la classe membre est une classe d'association qui présente les participants qui sont membres dans une équipe et qui est gérée par le responsable. Pour chaque équipe on a des cours

onlines et pour chaque cours online on aura une classe association conversation entre les membres de l'équipe. Encore on a la classe association publication, chaque utilisateur peut gérer des publications dans une équipe. L'utilisateur peut interagir en écrivant des commentaires dans une publication. Cette tâche présente par la classe association commentaire.

Enfin, la section des cours offline, chaque formation est une composition d'un ou plusieurs cours-offline-chapitre, chaque cours-offline-chapitre est une composition d'un ou plusieurs ressources et pour chaque chapitre on aura un seul quiz. Aussi on a la classe quiz, chaque quiz est composé par un ou plusieurs questions et chaque participant peut répondre à ces questions ce qui illustre dans la classe association reponseUtilisateur.

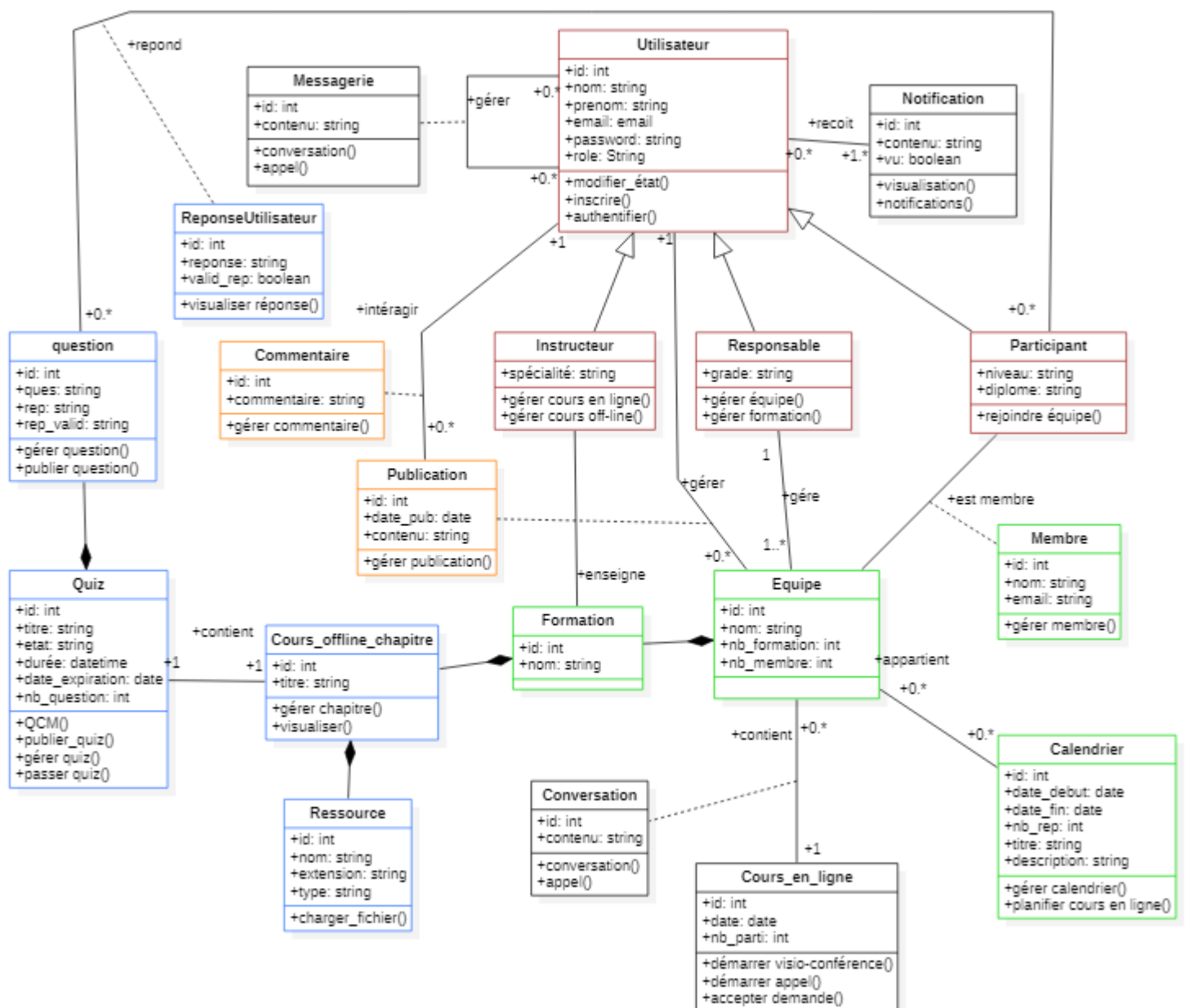


FIGURE 17 : DIAGRAMME DES CLASSE

VII. Conclusion :

Au cours de ce chapitre, nous tenons à présenter les missions des données par le « Backlog Product ». Pour ce faire, nous avons passé par l'analyse des besoins fonctionnels et non fonctionnels, la présentation des acteurs, le cas d'utilisation global, la planification du projet « Scrum », on a présenté l'architecture globale de projet avec l'environnement de développement et nous finissons par la présentation de la conception statique qui est le diagramme de classe. Dans le chapitre suivant nous présenterons les démarches de développement et de conception détaillé pour notre projet.

Chapitre 3 : Gestion d'accès d'utilisateurs et des publications (Sprint 1)

I. Introduction

L'authentification c'est une tâche primordiale pour garantir la sécurité de tous les utilisateurs et la répartition des tâches. Pour cela, dans ce chapitre, nous allons mettre en relief le premier sprint qui présent la gestion d'accès d'utilisateurs et des publications. Pour la première section on a décrit le backlog de sprint 1, en deuxième section on a décrit les diagrammes des cas d'utilisations, pour la troisième section on a proposé les solutions de diagramme de séquence, ensuite, nous allons décrit la réalisation et clôturé par une conclusion.

II. Le backlog du sprint 1 :

Id sprint	Thème	ID	Acteur	User Story	Priorité
1	Authentification	1.1	Utilisateur	En tant qu'utilisateur, je suis obligé de s'inscrire puis de s'authentifier à l'application.	Elevée
		1.2	Utilisateur	En tant qu'utilisateur, je peux quitter l'application.	Elevée
		1.3	Utilisateur	En tant qu'utilisateur, je veux que le mot de passe ne soit accepté que s'il respecte des règles bien définies.	Moyenne
	Gestion des équipes	1.4	Responsable	En tant que responsable d'établissement, je veux créer une équipe.	Elevée
		1.5	Utilisateur	En tant qu'utilisateur, je veux consulter une équipe.	Elevée
		1.6	Participant	En tant que participant, je veux chercher une équipe.	Elevée

		1.7	Participant	En tant que participant, je veux demander de rejoindre une équipe.	Elevée
	Gestion des membres	1.8	Responsable	En tant que responsable d'établissement, je veux ajouter un membre.	Elevée
		1.9	Responsable	En tant que responsable d'établissement, je veux accepter un membre.	Elevée
		1.10	Responsable	En tant que responsable d'établissement, je veux refuser un membre.	Elevée
		1.11	Responsable	En tant que responsable d'établissement, je veux mettre un membre exclu.	Elevée
		1.12	Responsable	En tant que responsable d'établissement, je veux consulter un membre.	Elevée
	Gestion des publications		Utilisateur	En tant qu'utilisateur, je veux consulter une publication.	Elevée
			Utilisateur	En tant qu'utilisateur, je veux ajouter une publication.	Elevée
			Utilisateur	En tant qu'utilisateur, je veux supprimer une publication.	Elevée
			Utilisateur	En tant qu'utilisateur, je veux consulter un commentaire.	Elevée
			Utilisateur	En tant qu'utilisateur, je veux ajouter un commentaire.	Elevée
			Utilisateur	En tant qu'utilisateur, je veux supprimer un commentaire.	Elevée

Tableau 5 : backlog du sprint1

III. Cadrage de besoins :

Au cours de cette étape, nous allons lister et détailler les besoins qui présenteront ce sprint.

1. Diagrammes des cas d'utilisation

1.1. Diagramme de cas d'utilisation « gérer équipe »

Ce sprint s'occupe de la gestion des équipes, il traite la création, la recherche et la consultation de chaque équipe. La gestion d'équipe nécessite l'authentification de responsable. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification, le responsable peut consulter équipe, créer équipe et cherche équipe. Ces relations sont réalisées par l'association extend. Le participant peut chercher équipe en demandant de rejoindre l'équipe, cette relation est réalisée par l'association extend entre chercher équipe et demande de rejoindre.

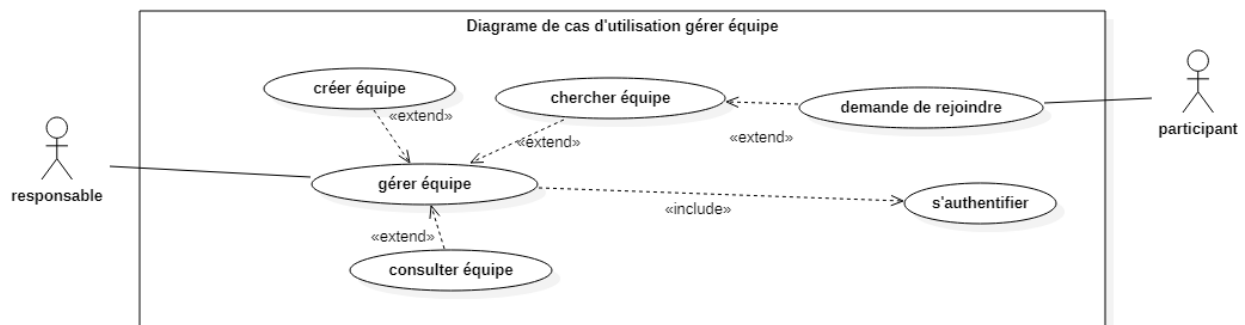


FIGURE 18: DIAGRAMME DE CAS D'UTILISATION DE " GERER EQUIPE "

1.2. Diagramme de cas d'utilisation « g rer membre »

Ce sprint s'occupe aussi de la gestion des membres de l' quipe, il traite l'ajout, l'acceptation, le refus et l'exclusion d'un membre. La gestion des membres nécessite l'authentification de responsable. Pour cela elle est li e par s'authentifier par une relation include. Apr s l'authentification, le responsable peut ajouter un membre, accepter un membre, refuser un membre et exclure un membre. Ces relations sont r alis es par l'association extend.

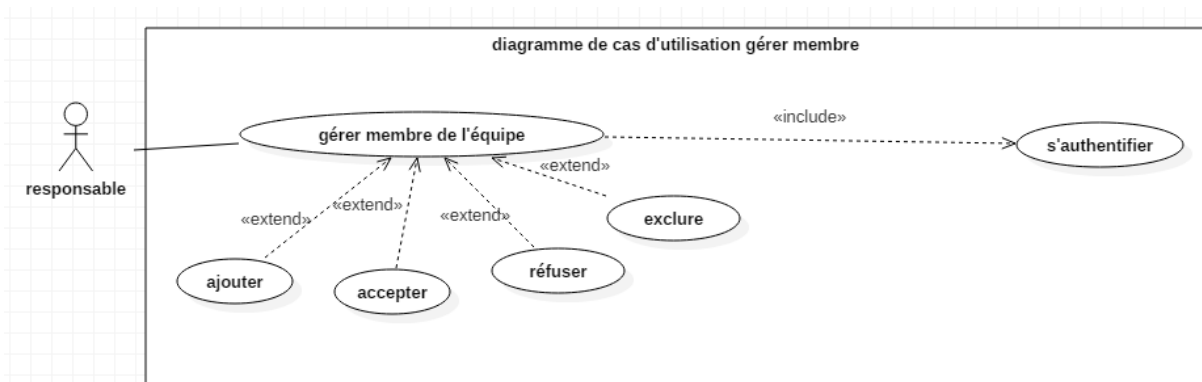


FIGURE 19: DIAGRAMME DE CAS D'UTILISATION DE "GERER MEMBRE"

1.3. Diagramme de cas d'utilisation « gérer publication »

Ce sprint s'occupe aussi de la gestion des publications et la gestion des commentaires, il traite la publication et la suppression d'une publication ou d'un commentaire. Les tâches de gestion des publications et des commentaires nécessitent une authentification de l'utilisateur. Pour cela elles sont tous liées par s'authentifier par une relation include. Après l'authentification, l'utilisateur peut publier, consulter et supprimer une publication. Ces relations sont réalisées par l'association extend. Après l'ajout d'une publication l'utilisateur peut ajouter, consulter et supprimer un commentaire. Ces relations sont réalisées par l'association extend.

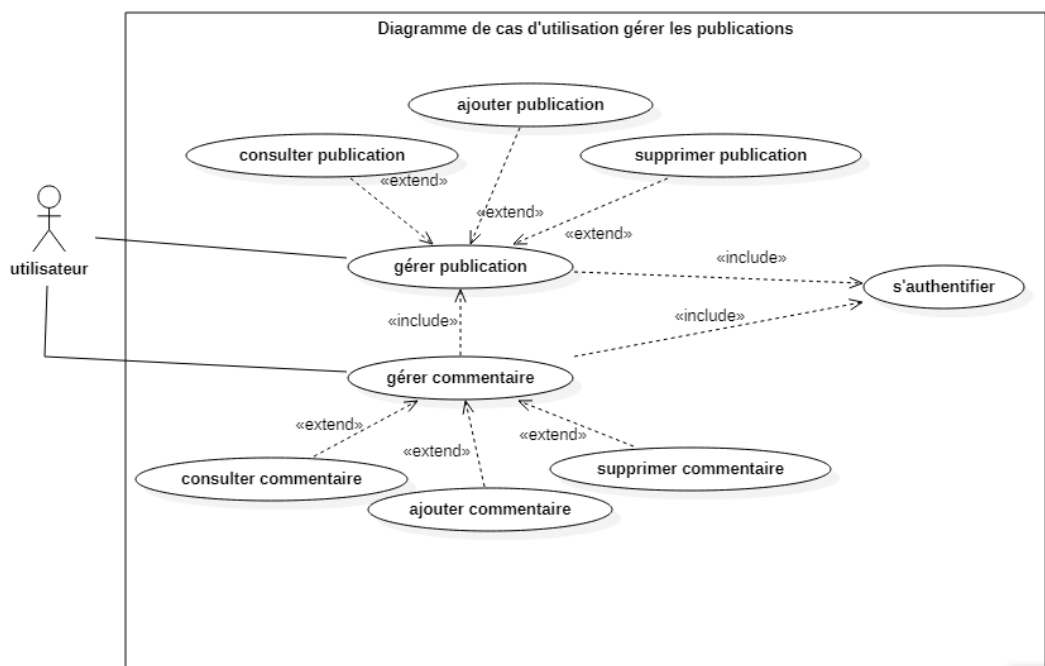


FIGURE 20: DIAGRAMME DE CAS D'UTILISATION DE "GERER PUBLICATION"

2. Description des cas d'utilisation

2.1. Description du cas d'utilisation S'authentifier

Ce tableau présente une description du cas d'utilisation "S'authentifier"

Titre	S'authentifier
Acteurs	Utilisateur (responsable, instructeur, participant)
Résumé	L'outil permet l'authentification de l'utilisateur
Description des enchainements	
Pré-conditions	L'utilisateur doit avoir un compte déjà créé.

	L'application doit être ouverte sur l'Interface d'authentification
Post-condition	L'utilisateur est connecté à son compte
Scénario nominal	
	L'utilisateur saisit son email. L'utilisateur saisit son mot de passe. L'utilisateur valide la connexion.
Scénario alternatif	
	E1 : Email d'utilisateur ou mot de passe non valide La page d'authentification est rechargée avec le message 'email d'utilisateur ou mot de passe non valide'.
IHM	Bouton : se connecter textView pour l'email d'utilisateur textView : mot de passe

TABLEAU 6: DESCRIPTION DE CAS D'UTILISATION AUTHENTIFICATION

2.2. Description du cas d'utilisation Gérer les équipes

Ce tableau présente une description du cas d'utilisation "créer équipe"

Titre	Créer équipe
Acteurs	Responsable
Résumé	L'outil permet la création des équipes.
Description des enchainements	
Pré-conditions	Responsable authentifié.
Post-condition	Equipe est créée
Scénario nominal	
	Le responsable clique Equipes dans le menu principal. Le responsable clique sur le bouton créer équipe. Le responsable remplit le champ nom d'équipe. Le responsable remplit le champ code d'équipe. Le responsable valide la création.
Scénario alternatif	
	E1 : Le responsable saisit des données non valides ou des champs.

	Le système affiche un message d'erreur.
IHM	Modal avec un formulaire Bouton : créer une équipe textView pour le nom de l'équipe textview : code équipe Bouton : créer

TABLEAU 7: DESCRIPTION DE CAS D'UTILISATION CREE EQUIPE

2.3. Description du cas d'utilisation Gérer les membres

Ce tableau présente une description du cas d'utilisation "Ajouter membre"

Titre	Ajouter membre
Acteurs	Responsable
Résumé	L'outil permet l'ajout des membres.
Description des enchainements	
Pré-conditions	Responsable authentifié. Equipe créé.
Post-condition	Membre est ajouté.
Scénario nominal	
	Le responsable clique « Equipes » dans le menu principal. Le responsable choisit une équipe. Le responsable clique sur l'icône « Membre » Le responsable remplit le champ email du membre. Le responsable valide l'ajout.
Scénario alternatif	
	E1 : Le responsable saisit un email non valide ou des champs vides. Le système affiche un message d'erreur.
IHM	textview : email du membre Bouton : ajouter

TABLEAU 8: DESCRIPTION DE CAS D'UTILISATION AJOUTER MEMBRE

IV. Conception

Dans cette partie, nous allons élucider la phase de conception, en exposant l'architecture technique et les scénarios à réaliser.

🚦 Scenarios

❖ Diagramme de séquence détaillé « s'authentifier »

Le diagramme de séquence fait partie des diagrammes d'interaction. Il permet de décrire les scénarios de réalisation des cas d'utilisation dans le système. Ce module traite la partie sécurité. On va détailler le processus d'authentification à travers le diagramme de séquence illustré dans la figure suivante. Si la réponse de la base des données est valide, la page d'accueil sera affichée, si non un message "email ou mot de passe invalide" sera affiché. Cette condition est expliquée par l'étiquette "alt" dans le diagramme de séquence.

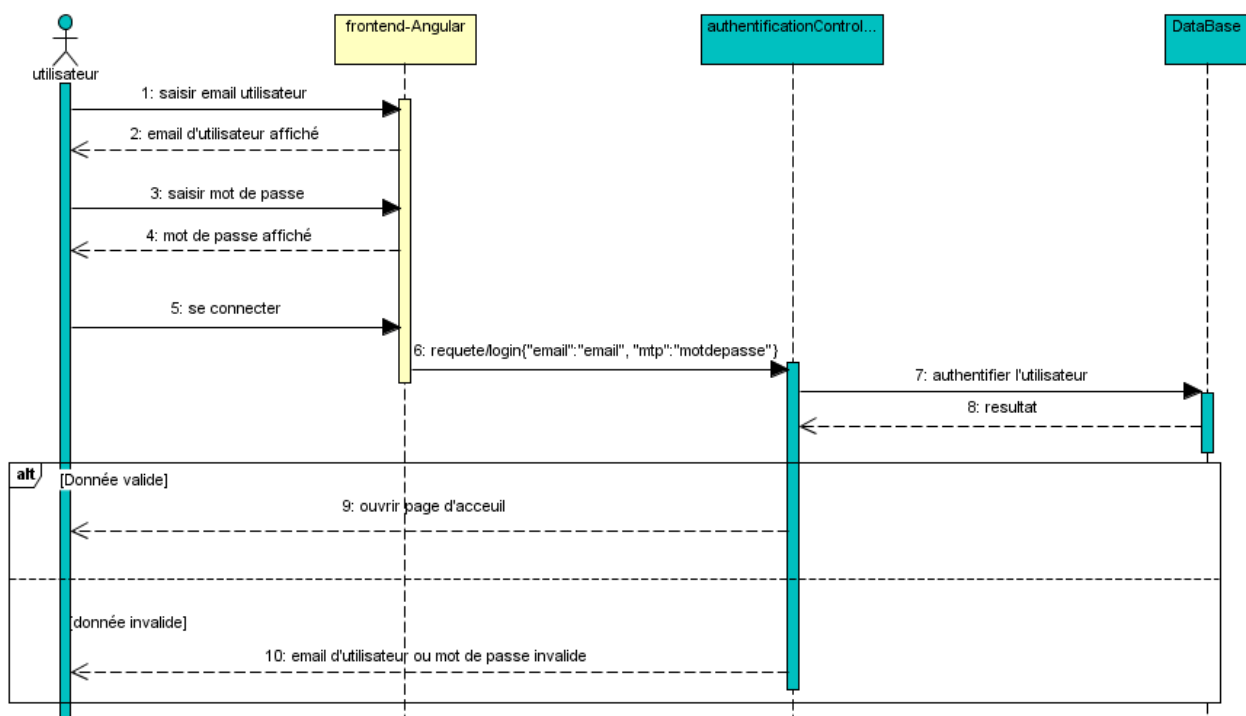


FIGURE 21: DIAGRAMME DE SEQUENCE S'AUTHENTIFIER

❖ Diagramme de séquence détaillé « créer équipe »

Dans cette partie nous prendrons en considération la partie « gestion d'équipe ». Nous présenterons le diagramme de séquence « créer équipe ». Dans ce diagramme, nous expliquerons la démarche dont un responsable doit suivre pour ajouter des équipes à la base de données des équipes. Dès que l'interface s'affiche, il cliquera sur le bouton créer équipe. Le modal de cette action s'affichera. Ensuite il remplit les champs et il valide. Si les champs sont

invalide un message d'erreur sera affiché. Sinon le 'front-end' (view) envoie une requête Post avec le chemin '/addEquipe' au 'EquipeController'. Le Controller enregistrera l'équipe dans la base des données. La figure suivante représente le diagramme de séquence détaillé pour la création d'une équipe.

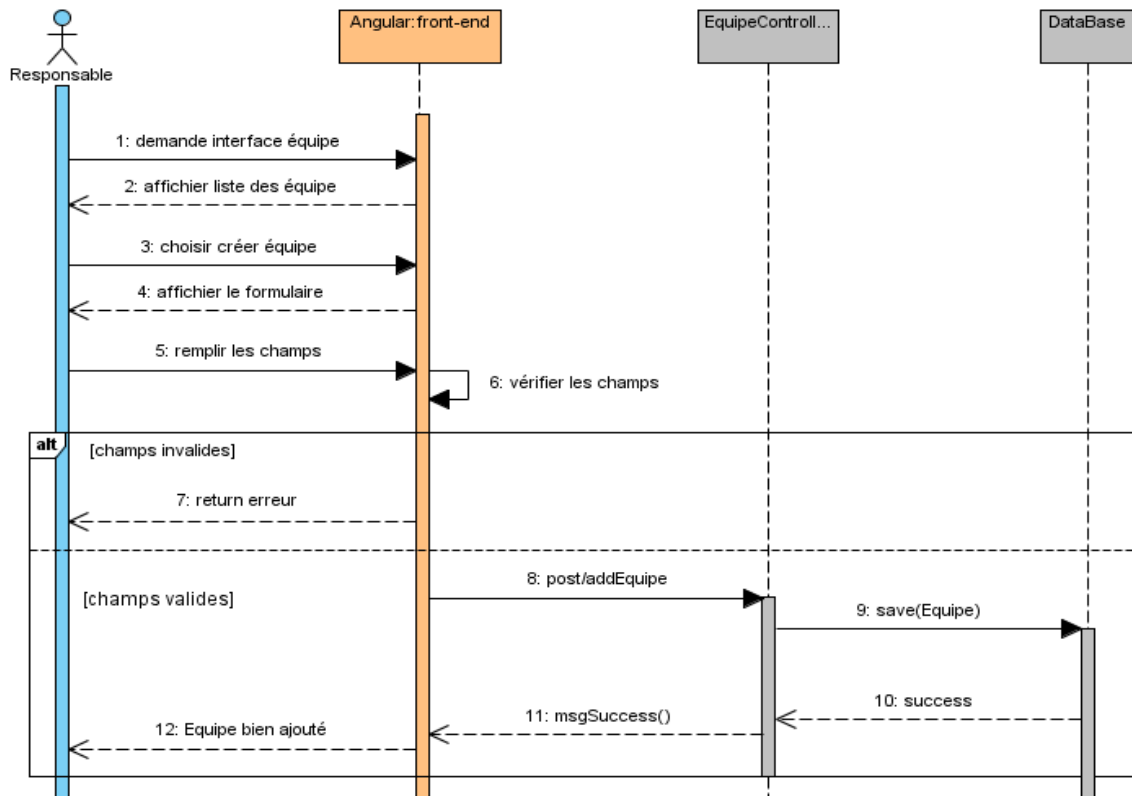


FIGURE 22: DIAGRAMME DE SEQUENCE AJOUTER EQUIPE

v. Réalisation

Dans cette partie, nous allons présenter quelques interfaces de l'application. Nous commençons par présenter la partie authentification : pour s'authentifier l'utilisateur doit saisir son email et son mot de passe, et clique sur « S'identifier ».

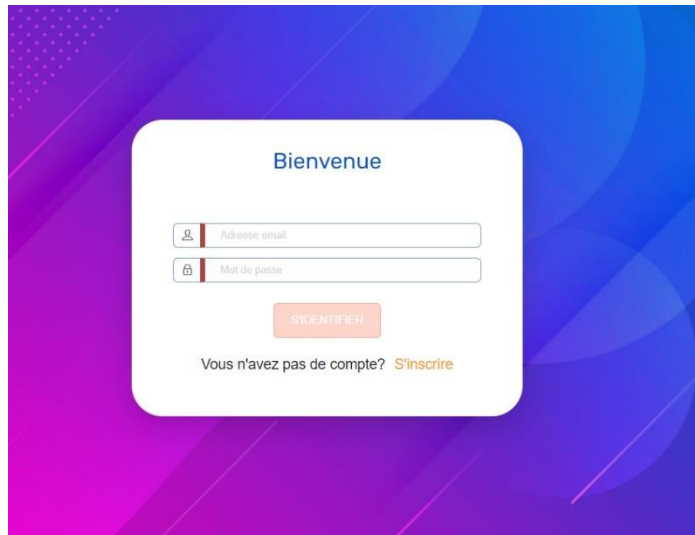


FIGURE 23: INTERFACE LOGIN

Dans cette application web on a trois types d'utilisateurs : responsable, instructeur et membre, donc on va présenter les interfaces de l'application dans les trois cas. Dans le cas où un responsable est connecté : Une fois le responsable est authentifié, une page d'accueil s'affiche. Le responsable a les permissions suivantes :

- Gérer les équipes.
- Gérer les membres.
- Gérer les cours



FIGURE 24: INTERFACE "ACCUEIL"

Passons maintenant à la gestion des équipes. En fait, afin de gérer les équipes, le responsable doit ouvrir la page : Gestion des équipes en cliquant sur « Equipe » dans le menu principal. Cette figure représente l'interface de gestion des équipes.

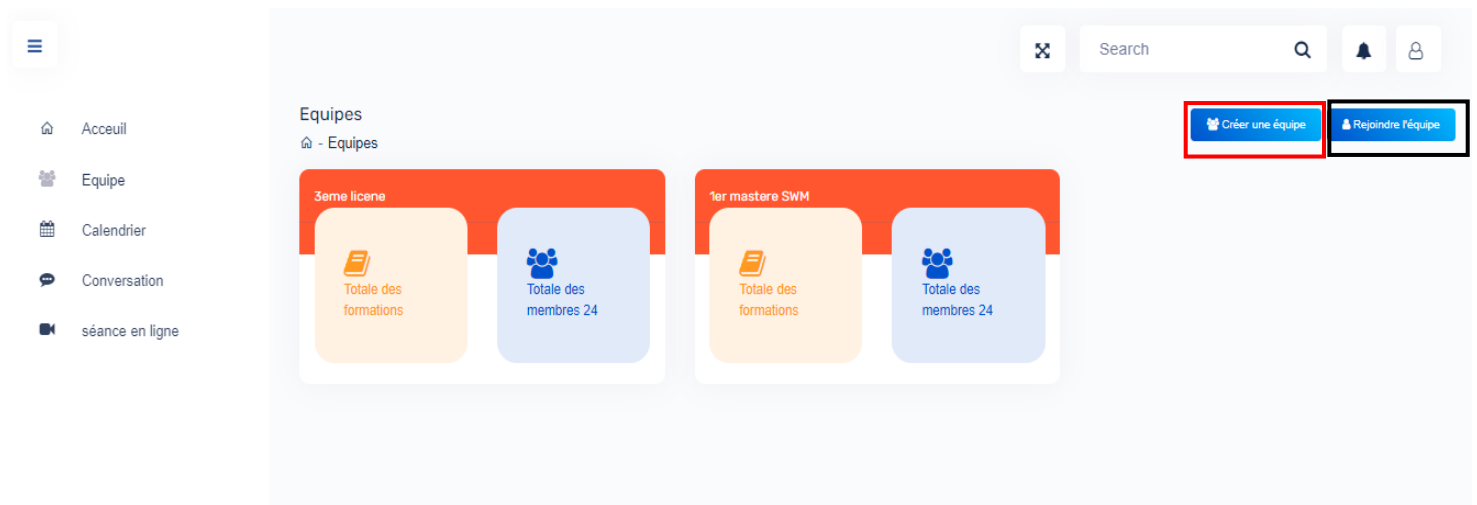


FIGURE 25:INTERFACE " GESTION DES EQUIPES"

Pour ajouter une équipe, le responsable doit cliquer sur le bouton « créer une équipe » qui est encadré en rouge dans la figure précédente. Le modal s'affiche et il devra ensuite remplir le formulaire. La figure suivante représente l'interface d'ajouter une équipe.



FIGURE 26: INTERFACE "AJOUTER EQUIPE"

Pour rejoindre une équipe, la participant doit cliquer sur le bouton « rejoindre équipe » qui est encadré en noir dans figure 25. Le modal s'affiche et il devra ensuite remplir le champ code équipe comme l'indique la figure suivante.

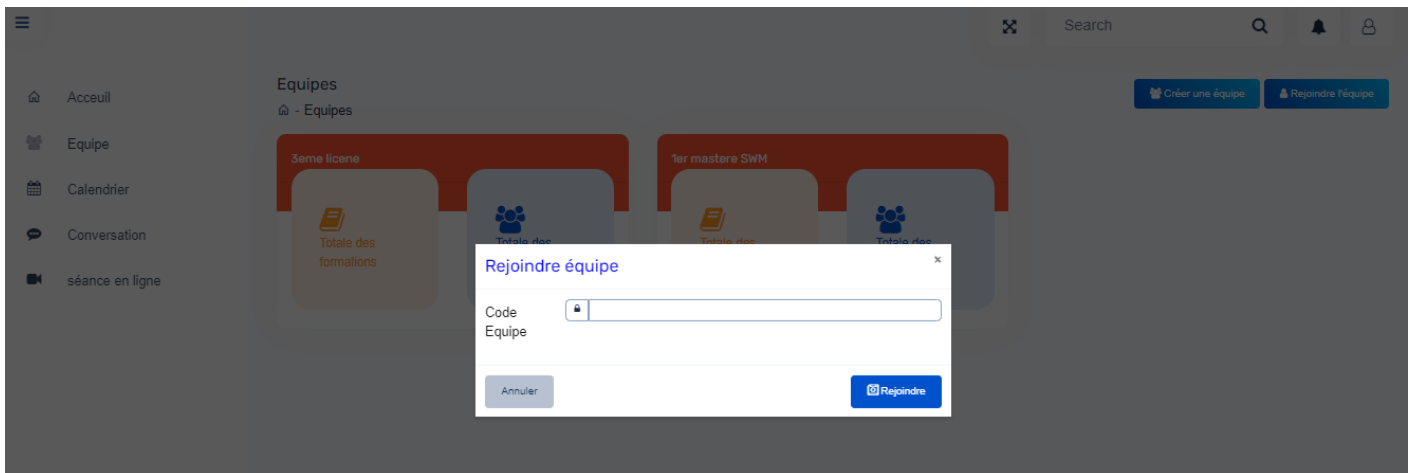


FIGURE 27: INTERFACE " REJOINDRE EQUIPE "

Passons maintenant à la gestion des membres. En fait, afin de gérer les membres, le responsable doit ouvrir la page : Gestion des membres en cliquant sur une équipe dans la page « Equipe » et on choisit la filière membre. Pour ajouter un membre le responsable doit remplir le champ adresse email d'un participant puis cliquer sur « Ajouter ».

Pour accepter une demande, le responsable clique sur le bouton « Accepter ».

Pour refuser une demande, le responsable clique sur le bouton « Refuser ».

Pour exclure un membre, le responsable clique sur le bouton « Exclure ».

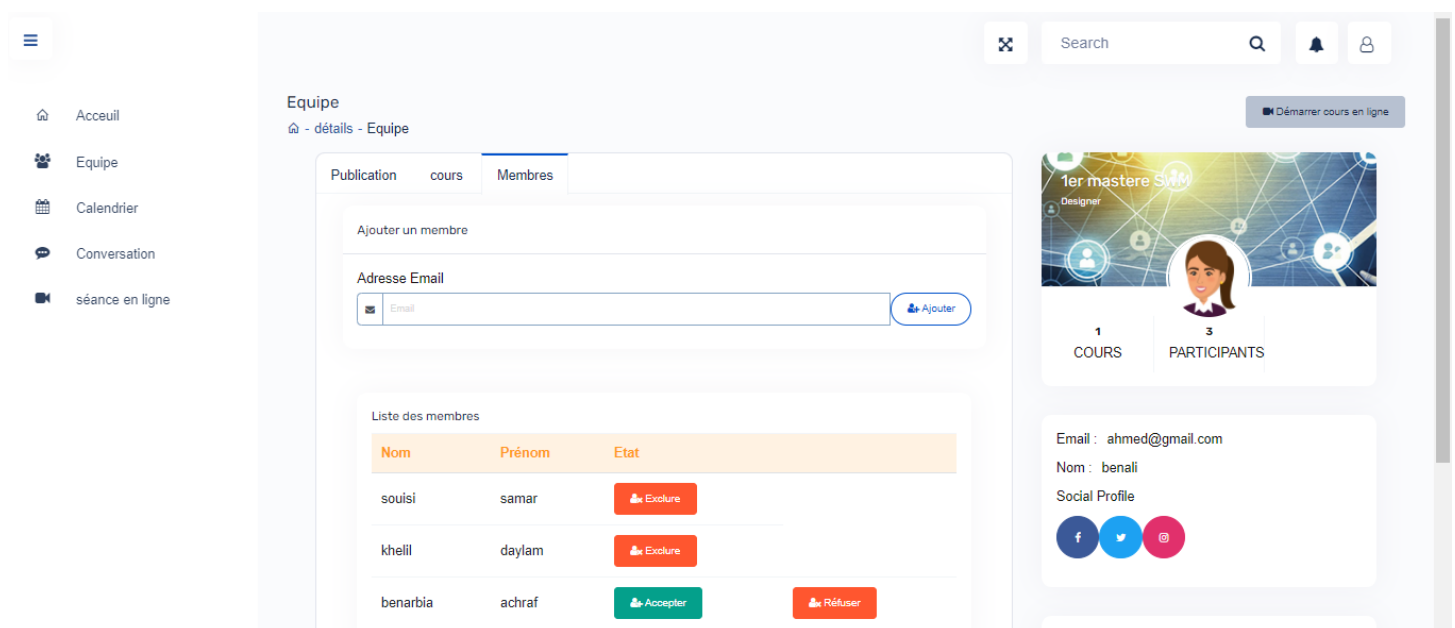


FIGURE 28: INTERFACE "GESTION DES MEMBRES"

Passons maintenant à la gestion des publications. En fait, afin de gérer les publications, l'utilisateur doit ouvrir la page : Gestion des publications en cliquant sur une équipe dans la page « Equipe » et on choisit la filière publication. Pour publier une publication, l'utilisateur doit remplir le champ « créer publication » et cliquer sur le bouton « Publier » comme l'indique la figure suivante dans le cadre en rouge. Après l'ajout d'une publication, chaque utilisateur de l'équipe peut publier des commentaires à cette publication, il remplit le champ de commentaire et clique sur l'icône ➡ comme elle est présente dans la figure suivante dans le cadre en vert. Chaque utilisateur peut supprimer une publication ou un commentaire en cliquant sur l'icône ✕ qui sont encadrés en noir dans la figure suivante.

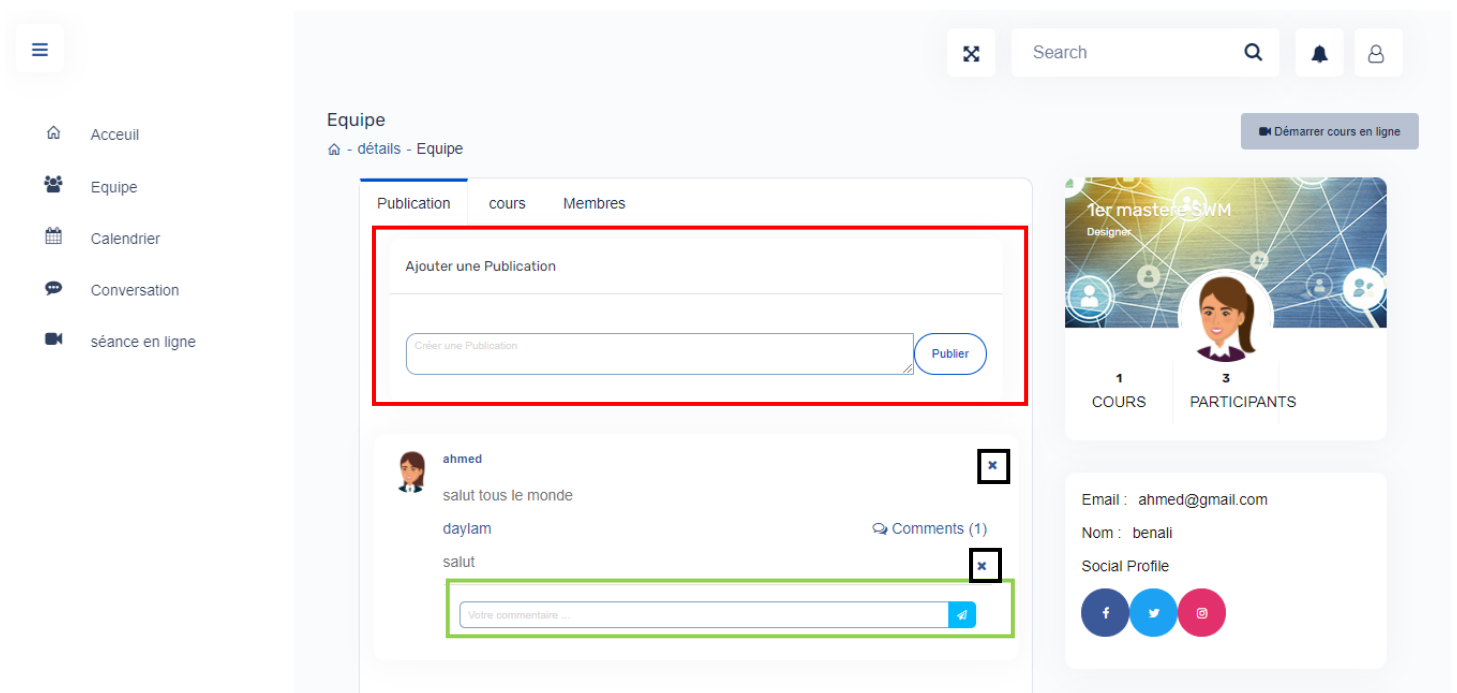


FIGURE 29:INTERFACE "GESTION DES PUBLICATIONS ET DES COMMENTAIRES"

VI. Conclusion :

Au cours de ce chapitre, nous avons réussi à mettre en place le sprint 1 de gestion d'accès d'utilisateurs et des publications. Nous avons passé par tout le cycle de développement : la spécification détaillée des besoins, la conception dynamique en exposant les différents diagrammes des cas d'utilisation, les diagrammes de séquence et enfin la réalisation où nous avons présenté les différentes interfaces. Dans le chapitre qui suit, notre effort sera consacré

pour produire un nouveau sprint qui abordera les fonctionnalités du deuxième sprint de ce projet.

Chapitre 4 : Gestion des cours offline (Sprint 2)

I. Introduction

Un deuxième sprint qui consiste à l'enseignement offline qui abordera la gestion des cours, des chapitre et d'évaluation. Pour cela ce sprint s'intéresse à gérer des cours offlines. Pour la première section on a décrit le backlog de sprint, en deuxième section on a décrit les diagrammes des cas d'utilisations, pour la troisième section on a proposé les solutions de diagramme de séquence, ensuite, nous allons décrit la réalisation et clôturé par une conclusion.

II. Le backlog du sprint 2 :

Id sprint	Thème	ID	Acteur	User Story	Priorité
2	Gestion de cours	1.1	Responsable	En tant que responsable d'établissement, je veux créer un cours.	Elevée
		1.2	Responsable	En tant que responsable d'établissement, je veux affecter un instructeur au cours.	Elevée
		1.3	Utilisateur	En tant qu'utilisateur, je veux consulter l'espace de cours.	Elevée
	Gestion de chapitre	1.4	Instructeur	En tant qu'instructeur, je veux ajouter un chapitre.	Elevée
		1.5	Instructeur	En tant qu'instructeur, je veux supprimer un chapitre.	Elevée
		1.6	Instructeur	En tant qu'instructeur, je veux publier une ressource.	Elevée
		1.7	Utilisateur	En tant qu'utilisateur, je veux ouvrir une ressource.	Elevée
		1.8	Instructeur	En tant qu'instructeur, je veux créer un quiz.	Elevée

	Gestion d'évaluation	1.9	Instructeur	En tant qu'instructeur, je veux publier un quiz.	Elevée
		1.10	Instructeur	En tant qu'instructeur, je veux modifier un quiz.	Elevée
		1.11	Instructeur	En tant qu'instructeur, je veux créer une question.	Elevée
		1.12	Instructeur	En tant qu'instructeur, je veux modifier une question.	Elevée
		1.13	Instructeur	En tant qu'instructeur, je veux supprimer une question.	Elevée
		1.14	Instructeur	En tant qu'instructeur, je veux consulter la liste des réponses de membre.	Elevée
		1.15	Participant	En tant que participant, je veux répondre à un quiz.	Elevée

TABLEAU 9 : BACKLOG DU SPRINT 2

III. Cadrage de besoins :

Au cours de cette étape, nous allons lister et détailler les besoins qui présenteront ce sprint.

1. Diagrammes des cas d'utilisation

1.1. Diagramme de cas d'utilisation « gérer cours »

Ce sprint s'occupe de la gestion des cours, il traite la création, la consultation d'espace de chaque cours et l'affectation d'un instructeur au cours. La gestion des cours nécessite une authentification de responsable. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification, le responsable peut consulter l'espace de cours et de créer un cours. Ces relations sont réalisées par l'association extend. Aussi il permet d'affecter un instructeur pour chaque cours, cette relation est réalisée par l'association extend entre créer cours et affecter un instructeur.

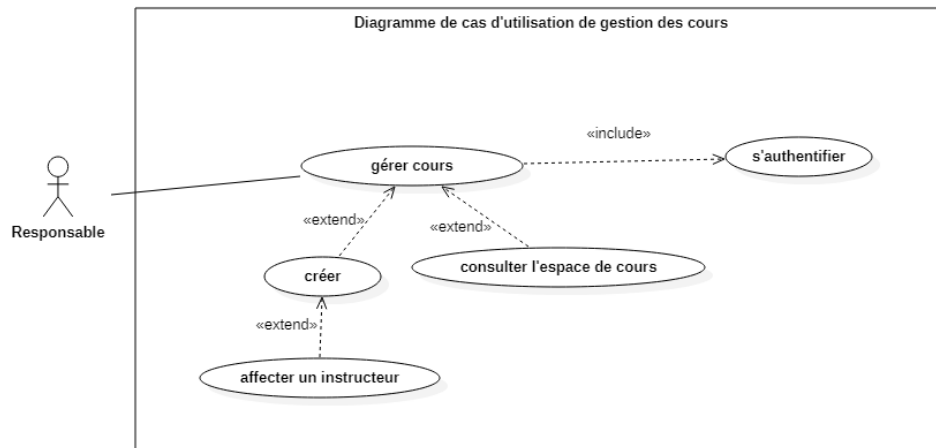


FIGURE 30:DIAGRAMME DE CAS D'UTILISATION "GESTION DES COURS"

1.2. Diagramme de cas d'utilisation « gérer chapitre »

Ce sprint s'occupe aussi de la gestion des chapitres, il traite la création, la publication, la suppression de chaque chapitre. La gestion des chapitres nécessite une authentification de l'instructeur. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification et l'affectation de l'instructeur, l'instructeur peut créer et supprimer chapitre. Ces relations sont réalisées par l'association extend. Aussi il permet de publier une ressource pour chaque chapitre, cette relation est réalisée par l'association extend entre créer chapitre et publier ressource.

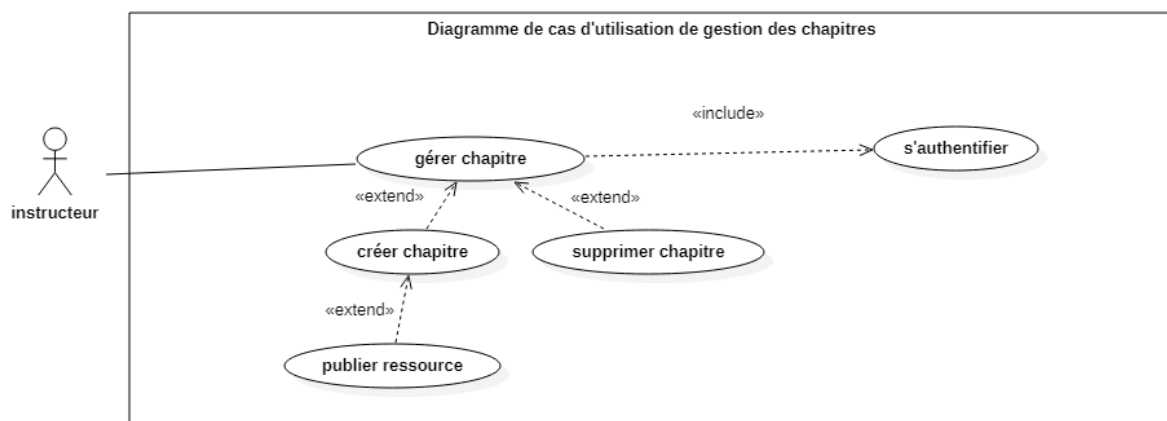


FIGURE 31:DIAGRAMME DE CAS D'UTILISATION "GESTION DES CHAPITRES"

1.3. Diagramme de cas d'utilisation « gérer évaluation »

Il s'occupe aussi de la gestion d'évaluation qui contient la gestion des quiz, il traite la création, la publication de quiz et la création des questions de chaque quiz, aussi il s'occupe de la consultation des réponses de chaque membre.

La gestion des quiz nécessite une authentification de l'instructeur. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification et la création d'un chapitre, l'instructeur peut créer et modifier un quiz. Ces relations sont réalisées par l'association extend. Après la création de quiz, l'instructeur peut publier quiz et gérer question, ces relations sont réalisées par l'association extend. L'instructeur peut créer, modifier et supprimer une question, ces relations sont réalisées par l'association extend avec gérer question. Notifier membre nécessite la publication d'un quiz, cette relation est réalisée par l'association include entre notifier membre et publier quiz. Après la publication de quiz, Le participant peut répondre aux quiz. Cette relation est réalisée par l'association extend. Après la réponse de participant, l'instructeur peut consulter les réponses, Cette relation est réalisée par l'association extend.

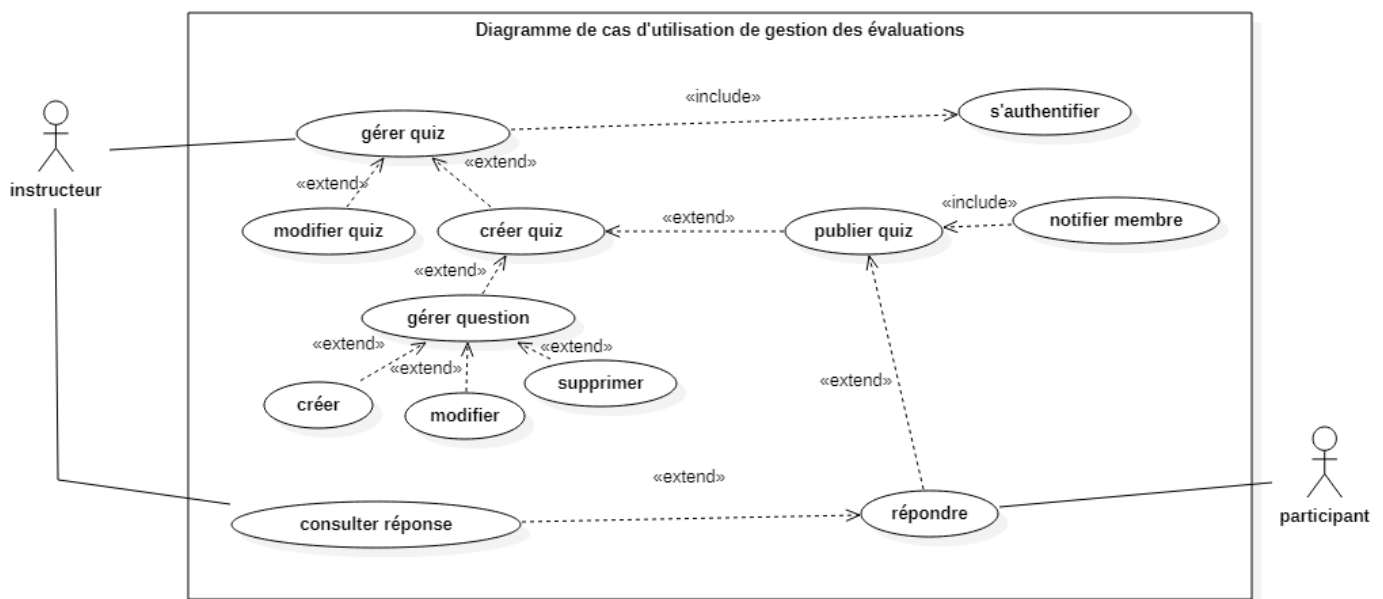


FIGURE 32:DIAGRAMME DE CAS D'UTILISATION "GESTION DES EVALUATIONS"

2. Description des cas d'utilisation

2.1. Description du cas d'utilisation « gestion de cours »

Ce tableau présente une description du cas d'utilisation "créer cours"

Titre	Créer cours
Acteurs	Responsable
Résumé	L'outil permet la création des cours.
Description des enchainements	
Pré-conditions	Responsable authentifié. Equipe créé.
Post-condition	Cours est crée
Scénario nominal	
	<p>Le responsable clique Equipes dans le menu principal.</p> <p>Le responsable choisit une équipe.</p> <p>Le responsable choisit la filière cours.</p> <p>Le responsable clique sur le bouton créer cours.</p> <p>Le responsable remplit le champ titre de cours.</p> <p>Le responsable remplit le champ email d'instructeur.</p> <p>Le responsable valide la création.</p>
Scénario alternatif	
	<p>E1 : Le responsable saisit des données non valides ou des champs vides ou email inexistant.</p> <p>Le système affiche un message d'erreur.</p>
IHM	<p>Bouton : créer un cours</p> <p>textView pour le titre de cours</p> <p>textview : email d'instructeur</p> <p>Deux Bouton : enregistrer et annuler</p>

TABLEAU 10:DESCRIPTION DE CAS D'UTILISATION CREER UN COURS

2.2. Description du cas d'utilisation Gérer les chapitres

Ce tableau présente une description du cas d'utilisation « Publier ressource »

Titre	Publier ressource
Acteurs	Instructeur
Résumé	L'outil permet la publication de ressource.
Description des enchainements	

Pré-conditions	Instructeur authentifié. Instructeur affecté au cours. Cours créé.
Post-condition	Ressource est publié.
Scénario nominal	
	L'instructeur clique cours dans le menu d'équipe. L'instructeur choisit un cours. L'instructeur clique sur le bouton plus « + » en bleu. L'instructeur clique sur le bouton browse. L'instructeur choisit un fichier. L'instructeur valide la publication.
Scénario alternatif	
	E1 : Le responsable saisit des données non valides ou des champs vides. Le système affiche un message d'erreur.
IHM	Modal avec un champ browser Bouton : plus (+) en bleu Deux Bouton : enregistrer et annuler

TABLEAU 11:DESCRIPTION DE CAS D'UTILISATION PUBLIER UNE RESSOURCE

2.3. Description du cas d'utilisation Gérer les évaluations

Ce tableau présente une description du cas d'utilisation "publier quiz"

Titre	Publier quiz
Acteurs	Instructeur
Résumé	L'outil permet la publication de quiz.
Description des enchainements	
Pré-conditions	Instructeur authentifié. Chapitre publié. Quiz créé.
Post-condition	Quiz est publié.
Scénario nominal	
	L'instructeur clique sur cours dans le menu d'équipe et choisit un cours.

	<p>L'instructeur choisit un chapitre et clique sur le bouton créer test.</p> <p>L'instructeur clique sur le bouton publier.</p> <p>L'instructeur remplit les champs date, heure et nombre de question.</p> <p>L'instructeur valide la publication.</p>
Scénario alternatif	
	<p>E1 : L'instructeur saisit des données non valides ou des champs vides.</p> <p>Le système affiche un message d'erreur.</p>
IHM	<p>Deux Bouton : publier</p> <p>Modal avec formulaire</p> <p>Calendrier : date et heure</p> <p>textView : nb question</p> <p>Bouton : annuler</p>

TABLEAU 12:DESCRIPTION DE CAS D'UTILISATION PUBLIER QUIZ

IV. Conception

Dans cette partie, nous allons élucider la phase de conception, en exposant l'architecture technique et les scenarios à réaliser.

Scenarios

❖ Diagramme de séquence détaillé « créer cours »

Dans cette partie, nous prendrons en considération la partie « gestion des cours ». Nous présenterons le diagramme de séquence « **créer cours** ». Dans ce diagramme nous expliquerons la démarche dont un responsable doit suivre pour ajouter des cours à la base de données des formations, dès que l'interface s'affiche, il cliquera sur le bouton créer cours. Le formulaire de cette action s'affichera. Ensuite il remplit les champs et il valide. Si les champs sont invalides un message d'erreur sera affiché. Sinon, le 'front-end' (view) envoie une requête Post avec le chemin '/addCours au 'CoursController'. Le Controller enregistrera le cours dans la base des données. La figure suivante représente le diagramme de séquence détaillée pour la création d'un cours.

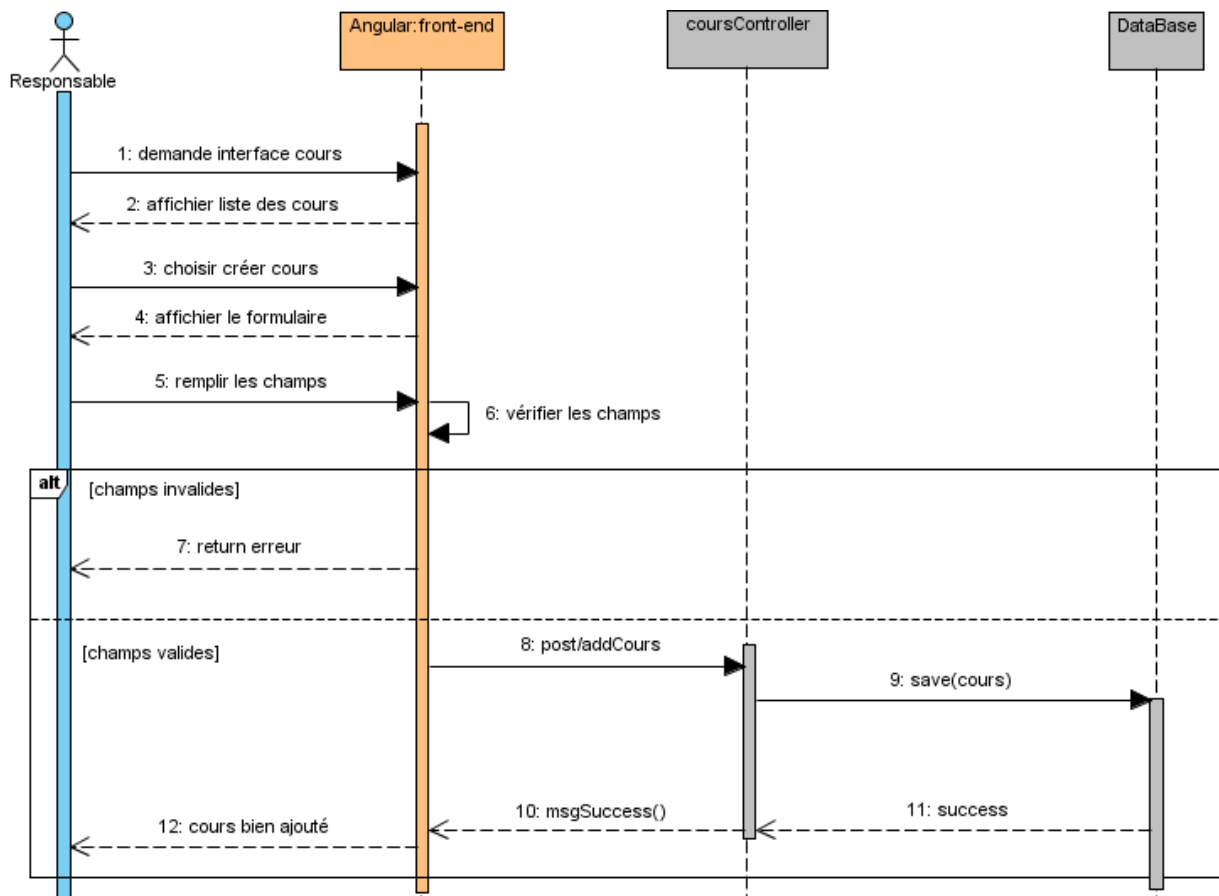


FIGURE 33:DIAGRAMME DE SEQUENCE CREER COURS

❖ Diagramme de séquence détaillé « publier quiz »

Dans cette partie, nous prendrons aussi en considération la partie « gestion de chapitre ». Nous présenterons le diagramme de séquence « **publier Quiz** ». Dans ce diagramme, nous expliquerons la démarche dont un instructeur doit suivre pour ajouter des quiz à la base de données des quiz, dès que l'interface s'affiche, il cliquera sur le bouton publier. Le modal de cette action s'affichera. Ensuite, il remplit les champs et il valide. Si les champs sont invalides un message d'erreur sera affiché. Sinon, le 'front-end' (view) envoie une requête Put avec le chemin '/publierQuiz au 'QuizController'. Le Controller enregistrera le quiz dans la base des données. La figure11 représente le diagramme de séquence détaillé pour la publication d'un quiz.

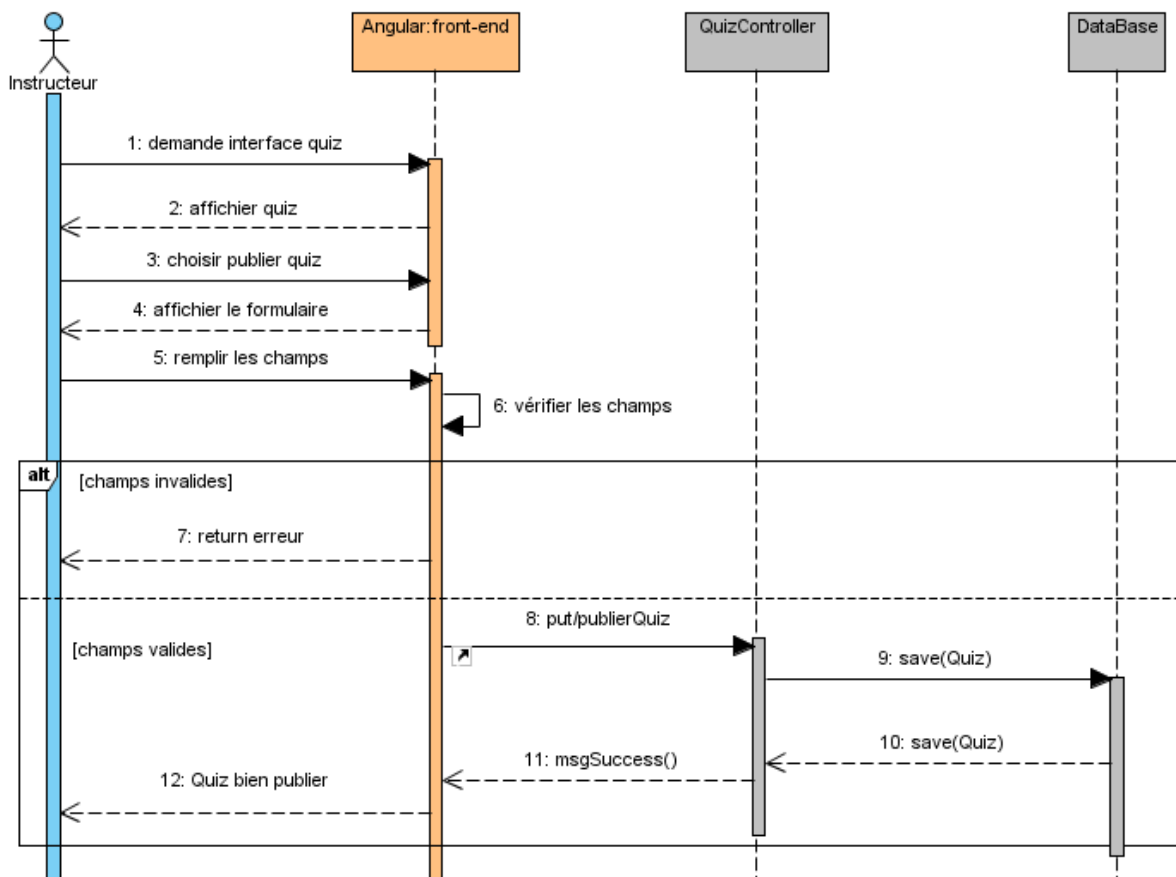


FIGURE 34:DIGRAMME DE SEQUENCE PUBLIER QUIZ

❖ Diagramme de séquence détaillé « supprimer chapitre »

Dans cette partie, nous prendrons aussi en considération la partie « gestion de chapitre ». Nous présenterons le diagramme de séquence « **supprimer chapitre** ». L'instructeur clique sur supprimer pour la suppression d'un chapitre. Si la suppression est invalide un message d'erreur sera affiché. Sinon, le 'front-end' (view) envoie une requête Delete avec le chemin '/chapitre/delete/idchapitre' au 'ChapitreController'. Le Controller enregistrera les modifications dans la base des données. La figure suivante représente le diagramme de séquence détaillé pour la suppression d'un chapitre.

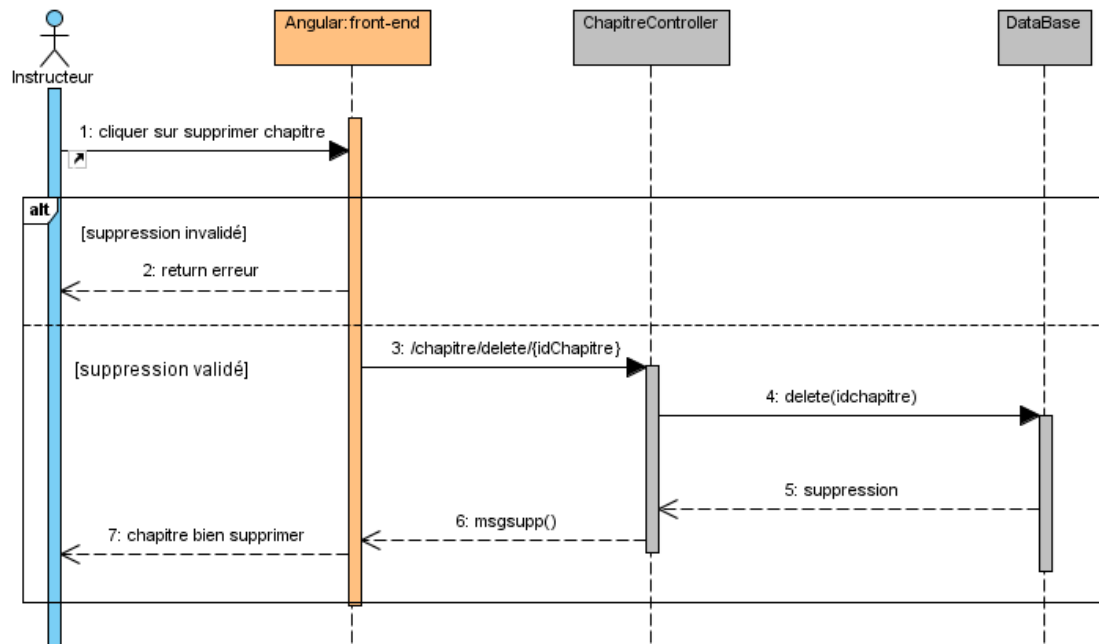


FIGURE 35:DIAGRAMME DE SEQUENCE SUPPRIMER CHAPITRE

V. Réalisation

Cette partie présentera les différentes interfaces de la gestion des cours-offlines.

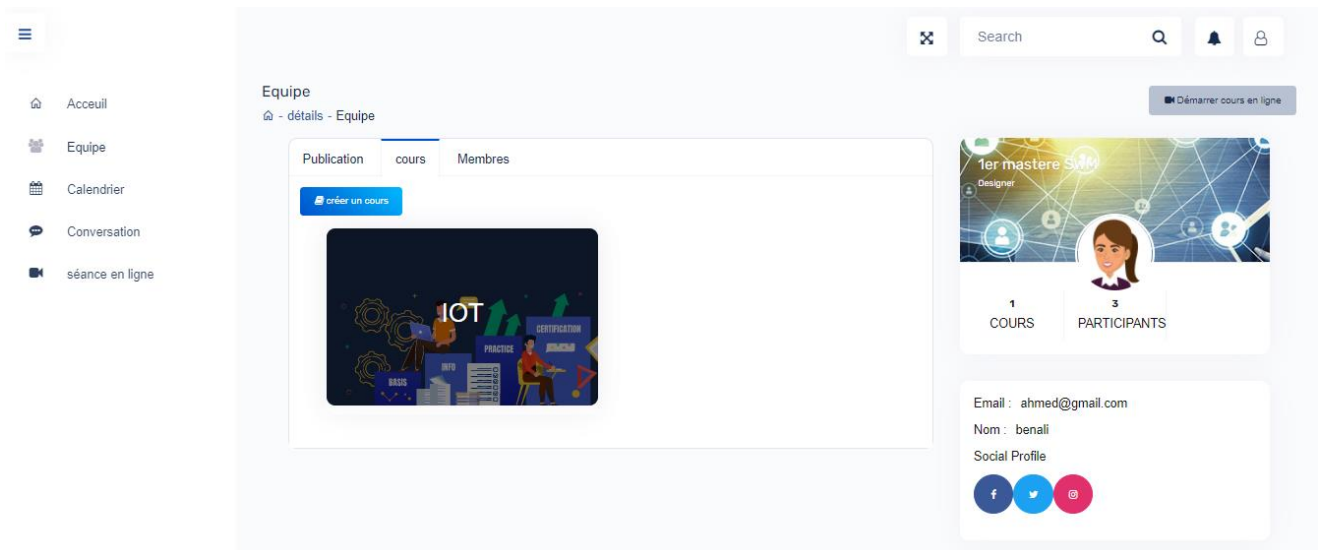


FIGURE 36:INTERFACE "GESTION DES COURS"

Pour ajouter un cours, le responsable doit cliquer sur le bouton « créer un cours ». Cette figure représente l'interface de la création d'un cours.

The screenshot shows the 'Création d'un cours' (Course Creation) interface. On the left is a sidebar with navigation links: Accueil, Equipe, Calendrier, Conversation, and séance en ligne. The main content area is titled 'Création d'un cours' and shows a breadcrumb 'Formulaire - Création d'un cours'. Below this is a form with fields for 'Cours' (Title), 'Email d'instructeur', and a 'Remember me' checkbox. At the bottom are 'Annuler' and 'Enregistrer' buttons.

FIGURE 37:INTERFACE " AJOUTER COURS"

Passons maintenant à la gestion des chapitres. Pour afficher le contenu de cours, l'utilisateur doit cliquer sur le cours. Cette interface est affichée pour le responsable, il permet de voir la liste des chapitres et leurs ressources.

The screenshot shows the 'Contenu de Cours' (Course Content) interface for the course 'IOT'. On the left is a sidebar with navigation links: Accueil, Equipe, Calendrier, Conversation, and séance en ligne. The main content area is titled 'Cours : IOT' and shows a breadcrumb 'Contenu de Cours - Cours'. Below this is a tabbed interface with 'Contenus' selected. It displays a 'Liste des chapitres' (List of chapters) with a table showing 'Titre de chapitre' and 'Les ressources'. The first row shows 'chapitre 1 :introduction' and a video thumbnail.

FIGURE 38:INTERFACE "CONTENU COURS POUR RESPONSABLE"

Le participant peut consulter les chapitres et cliquer sur le bouton en bleu « passer test » pour accéder à l'interface de test.

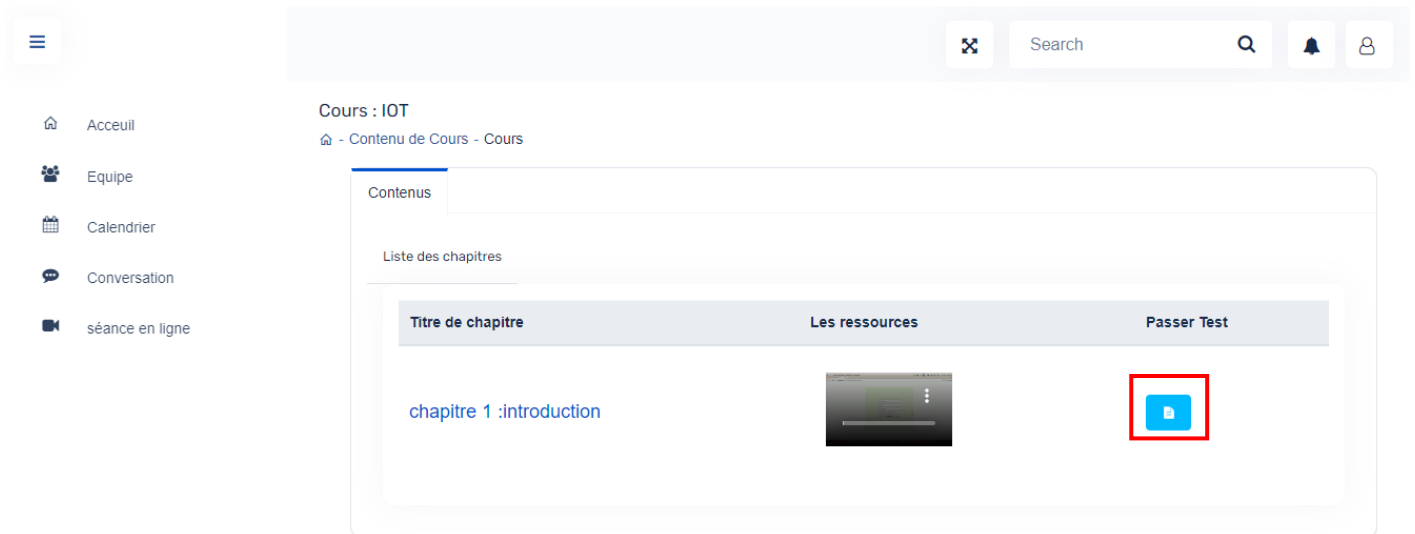


FIGURE 39:INTERFACE " CONTENU COURS POUR PARTICIPANT"

Cette interface est accessible pour l'instructeur. Pour ajouter un chapitre, l'instructeur doit remplir le champ titre et cliquer sur le bouton « Ajouter » comme l'indique la figure suivante.

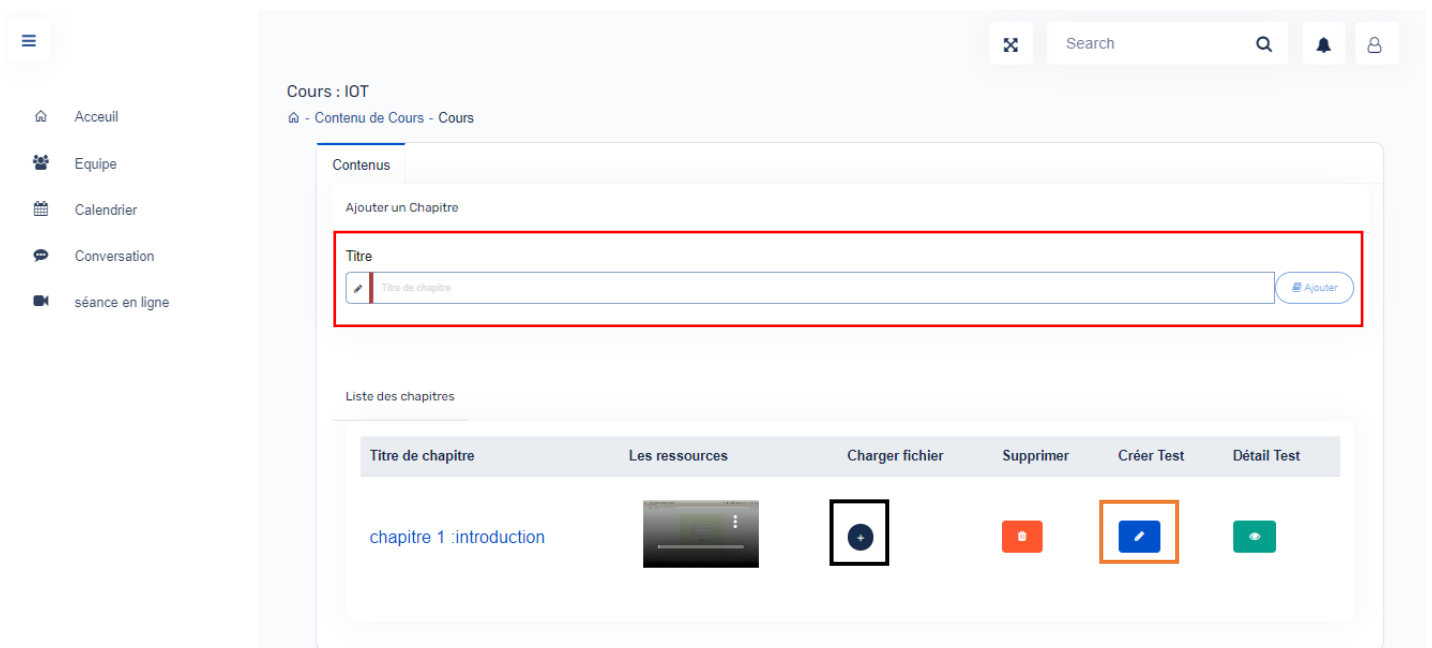


FIGURE 40:INTERFACE "GESTION DE CHAPITRE"

Pour publier un chapitre, l'instructeur doit charger un fichier, il doit cliquer sur le bouton « + » qui est encadré en noir comme l'indique la figure précédente. Cette figure représente l'interface de chargement d'un fichier.

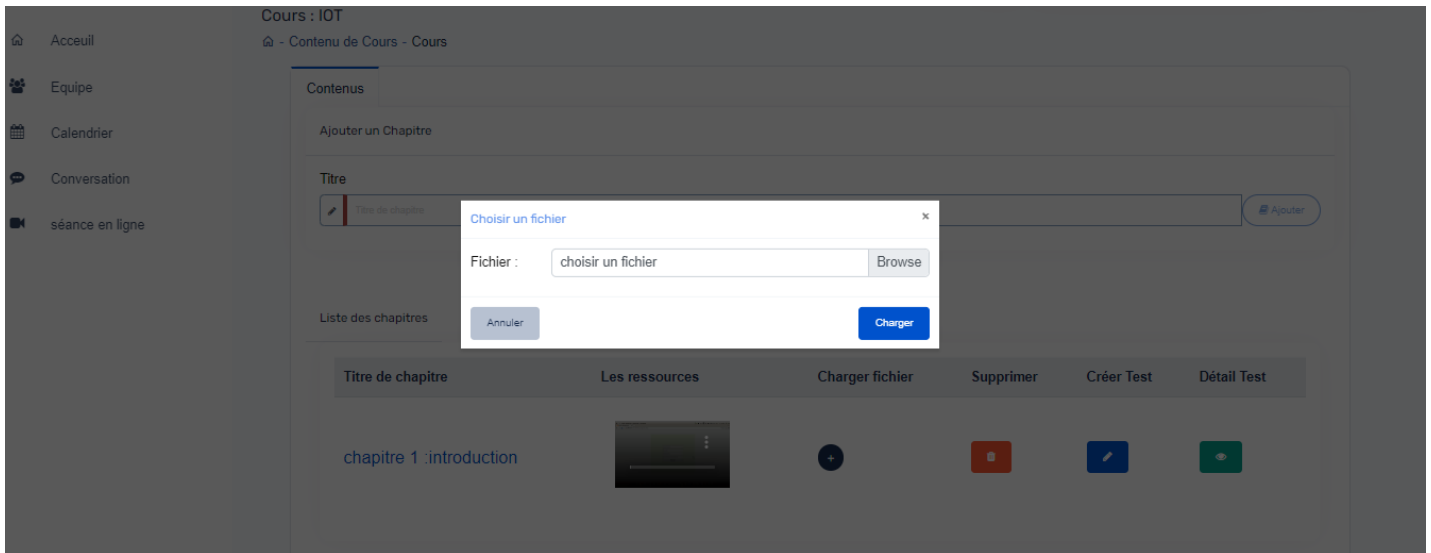


FIGURE 41:INTERFACE "CHARGER FICHIER"

Passons maintenant à la gestion des évaluations. Pour créer un test, l'instructeur doit cliquer sur le bouton « créer test » est encadré en orangé comme l'indique la figure 40. Cette figure représente l'interface de gestion de quiz.

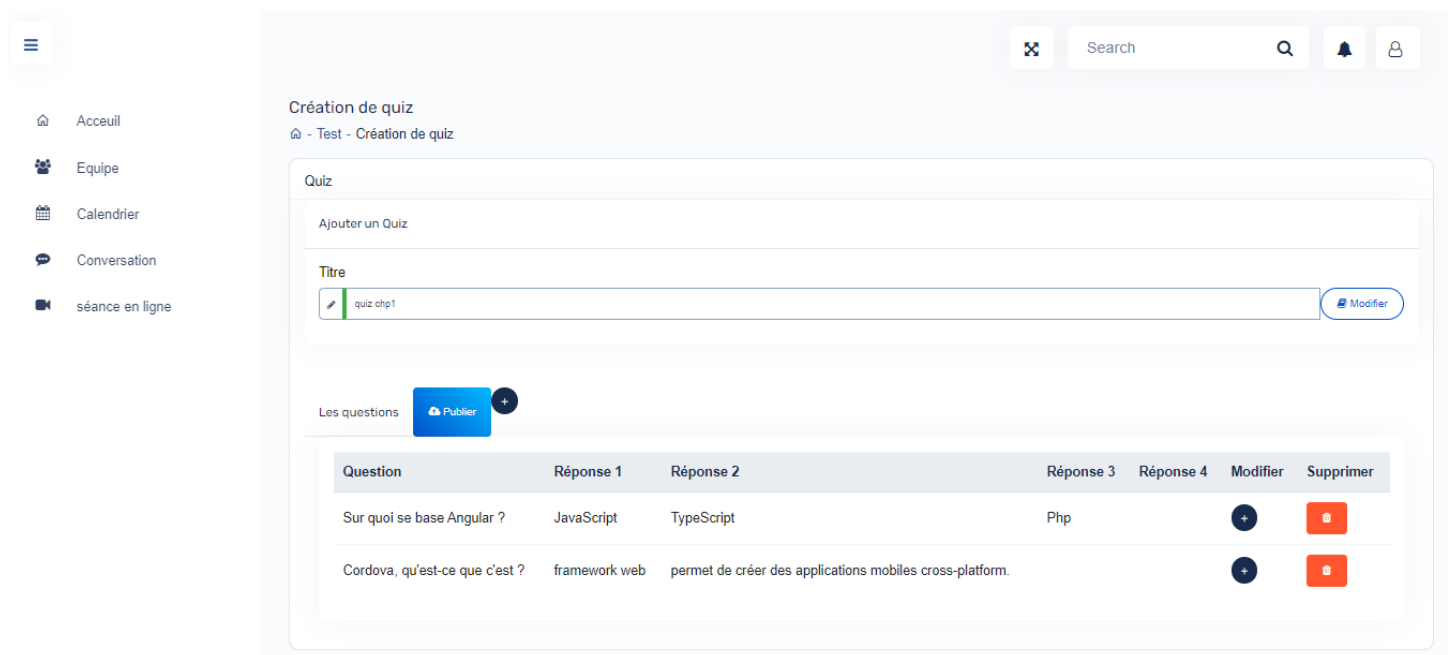


FIGURE 42 : INTERFACE GESTION DE QUIZ

Pour ajouter un quiz, l'instructeur doit remplir le champ titre de quiz puis cliquer sur le bouton « Ajouter ». Après l'ajout, il peut modifier le titre de ce quiz comme l'indique la figure suivante dans le cadre en orangé.

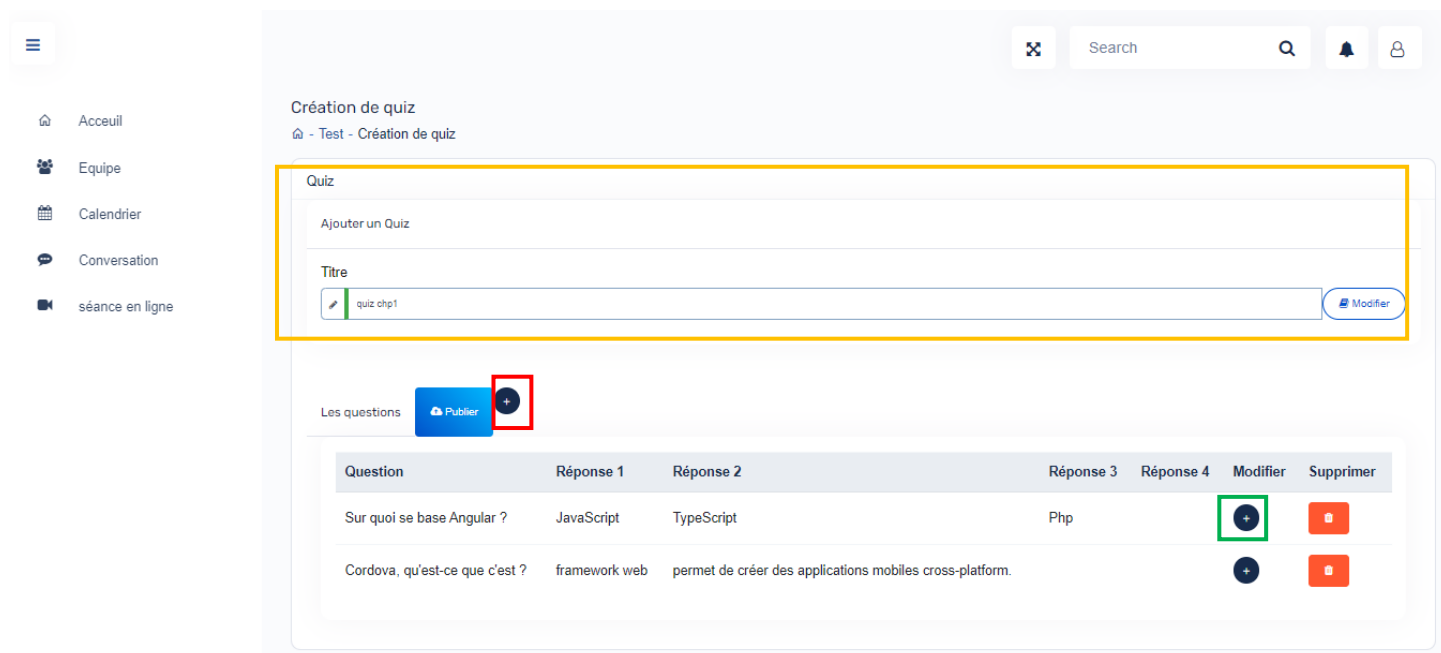


FIGURE 43:INTERFACE "GESTION DES QUIZ"

Pour ajouter une question a ce quiz, l'instructeur doit cliquer sur le bouton « + » qui encadré en rouge, il doit remplir le formulaire et cocher la bonne réponse. Cette figure représente l'interface ajouter question.

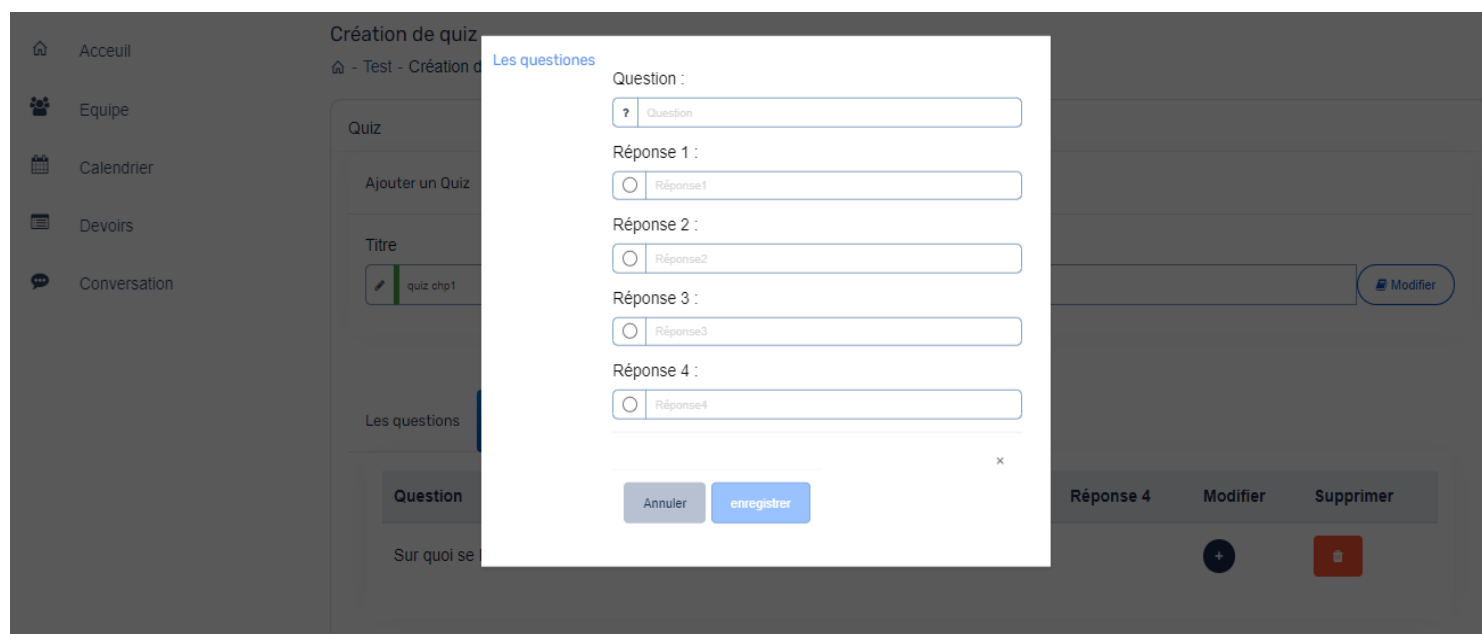


FIGURE 44:INTERFACE "AJOUTER QUESTION"

Pour publier un quiz, l'instructeur clique sur le bouton « Publier » en bleu et il doit remplir les champs. Cette figure représente l'interface de publier quiz.

The screenshot shows the 'Création de quiz' (Quiz Creation) interface. A modal titled 'Détails' is open, allowing the user to publish the quiz. The modal contains the following fields:

- Date:** A date picker showing 'jj/mm/aaaa'.
- Heures:** A time picker showing 'hh:mm'.
- Nombre de question:** A numeric input field.

At the bottom of the modal are two buttons: 'Annuler' (Cancel) and 'Publier' (Publish). The background interface shows a sidebar with navigation options (Accueil, Equipe, Calendrier, Devoirs, Conversation) and a main area with a 'Quiz' section containing a table of questions.

Question	Réponse 1	Réponse 2	Réponse 3	Réponse 4	Modifier	Supprimer
Sur quoi se base Angular ?	JavaScript	TypeScript	Php			

FIGURE 45:INTERFACE "PUBLIER QUIZ"

Pour modifier une question, l'instructeur doit cliquer sur le bouton « + » qui est encadré en vert. Une modal s'affichera avec les informations. Cette interface représente l'interface modifier question.

The screenshot shows the 'Création de quiz' interface with a modal titled 'Les questions' open for editing a question. The modal contains the following fields:

- Question :** A text input field with a question mark icon, containing the text 'Sur quoi se base Angular ?'.
- Réponse 1 :** A radio button followed by a text input field containing 'JavaScript'.
- Réponse 2 :** A radio button followed by a text input field containing 'TypeScript'.
- Réponse 3 :** A radio button followed by a text input field containing 'Php'.
- Réponse 4 :** A radio button followed by a text input field containing 'Réponse4'.

At the bottom of the modal are two buttons: 'Annuler' (Cancel) and 'enregistrer' (Save). The background interface shows the same sidebar and a table of questions.

Question	Réponse 1	Réponse 2	Réponse 3	Réponse 4	Modifier	Supprimer
Sur quoi se base Angular ?	JavaScript	TypeScript	Php			

FIGURE 46:INTERFACE "MODIFIER QUESTION"

Après la publication du test, le membre peut passer ce test, il doit cliquer sur le bouton « passer test » qui est encadré en rouge dans la figure 39. Cette figure représente l'interface passer test.



FIGURE 47:INTERFACE " PASSER TEST"

Pour démarrer le test, le membre doit cliquer sur le bouton « Démarrer la minuterie » qui est encadré en rouge dans la figure précédente. Le participant doit choisir une réponse et cliquer sur le bouton suivant pour terminer le test dans une durée précis. La figure suivante représente l'interface de test.

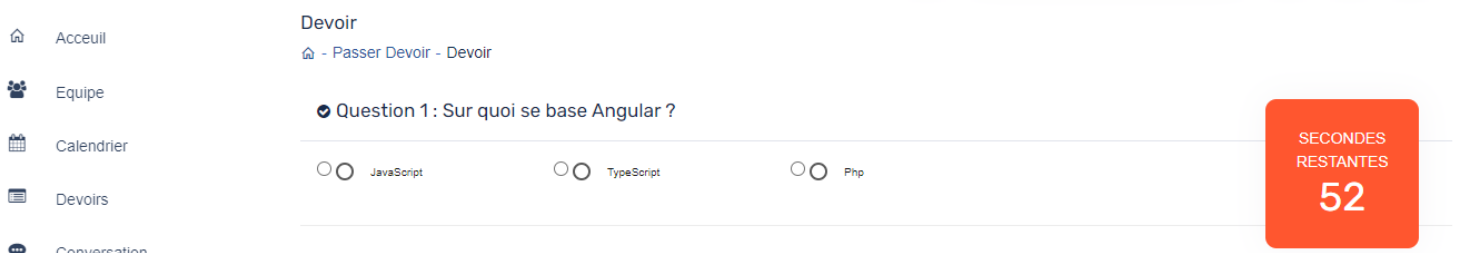


FIGURE 48:INTERFACE "TEST"

Cette figure représente l'interface de nombre des réponses correct de test pour le participant lorsqu'il termine le test.

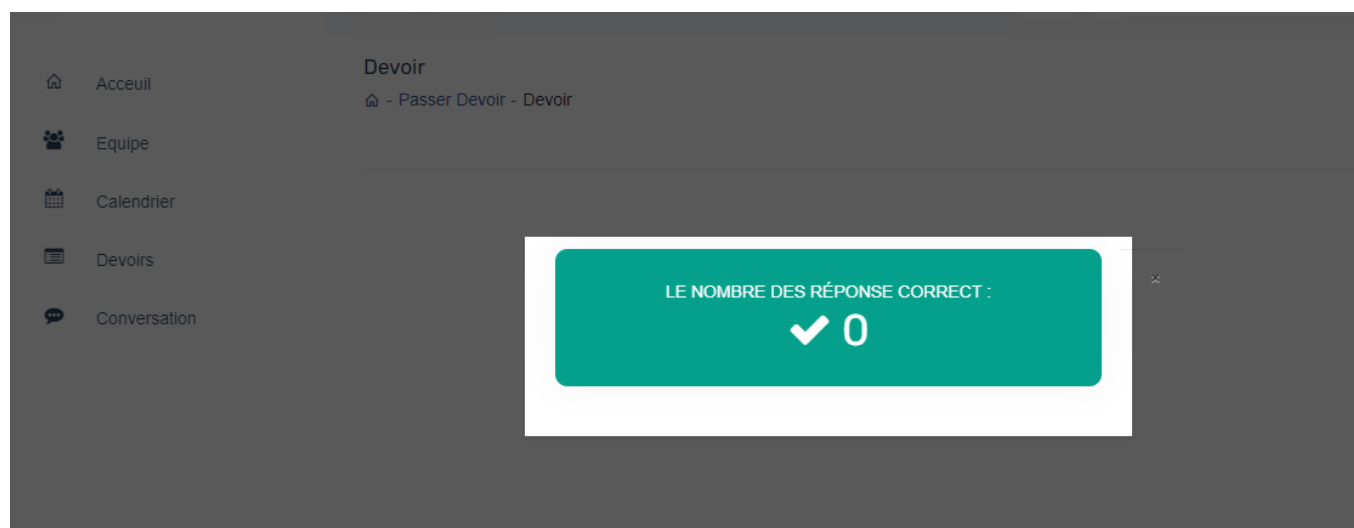


FIGURE 49:INTERFACE " REPONSE CORRECT"

Si le chapitre ne comprend pas de test, le membre reçoive ce message. La figure suivante représente l'interface lorsqu'il ne contient pas de test.

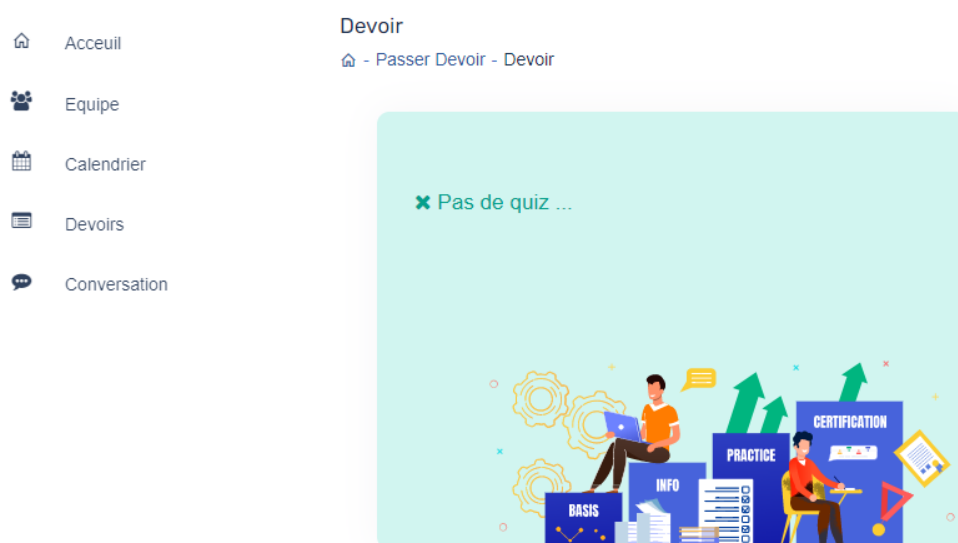


FIGURE 50:INTERFACE " TEST"

VI. Conclusion

Au terme de ce chapitre, nous avons traité le deuxième sprint avec succès qui a pour objet de gérer les cours-offlines par la répartition par acteur, la conception et la réalisation des interfaces pour ce sprint. A ce stade il reste le module de gestion des cours-online, qui fera l'objectif du suivant sprint

Chapitre 5 : Gestion des cours online (sprint 3)

I. Introduction

Ce dernier sprint consiste à l'enseignement online qui est un type d'enseignement bien d'actualité qui contient la gestion de visio-conférence, les notifications et la messagerie instantanée. Pour cela ce sprint s'intéresse à gérer des cours online. Pour la première section on a décrit le backlog de sprint, en deuxième section on a décrit les diagrammes des cas d'utilisations, pour la troisième section on a proposé les solutions de diagramme de séquence, ensuite, nous allons décrire la réalisation et clôturer par une conclusion.

II. Le backlog du sprint 3 :

Id sprint	Thème	ID	Acteur	User Story	Priorité
3	Gestion du calendrier et visio-conférence	1.1	Instructeur	En tant qu'instructeur, je veux planifier une visio-conférence dans le calendrier.	Elevée
		1.2	Instructeur	En tant qu'instructeur, je veux modifier la planification de visio-conférence sur le calendrier.	Elevée
		1.3	Instructeur	En tant qu'instructeur, je veux démarrer une visio-conférence.	Elevée
		1.4	Instructeur	En tant qu'instructeur, je veux consulter une visio-conférence.	Moyenne
		1.5	Instructeur	En tant qu'instructeur, je veux annuler une visio-conférence	Moyenne
		1.6	Participant	En tant que participant, je veux accepter une demande de visio-conférence.	Elevée
		1.7	Participant	En tant que participant, je veux annuler une demande de visio-conférence.	Moyenne

	Gestion de messagerie	1.8	Utilisateur	En tant qu'utilisateur, je veux sélectionner un récepteur.	Elevée
		1.9	Utilisateur	En tant qu'utilisateur, je veux envoyer un message.	Elevée
		1.10	Utilisateur	En tant qu'utilisateur, je veux consulter la liste des contacts.	Elevée
		1.11	Utilisateur	En tant qu'utilisateur, je veux passer un appel.	Elevée
	Gestion de notification	1.12	Instructeur	En tant qu'instructeur, je reçois une notification pour l'affectation à un cours.	Elevée
		1.13	Participant	En tant que participant, je reçois une notification pour la planification d'une visio-conférence.	Elevée
		1.14	Participant	En tant que participant, je reçois une notification pour le démarrage d'une séance de visio-conférence.	Elevée
		1.15	Participant	En tant que participant, je reçois une notification pour la publication d'un quiz.	Elevée

TABLEAU 13: BACKLOG DU SPRINT3

III. Cadrage de besoins :

Au cours de cette étape, nous allons lister et détailler les besoins qui présenteront ce sprint.

1. Diagrammes des cas d'utilisation

1.1. Diagramme de cas d'utilisation « gérer calendrier et visio-conférence »

Ce sprint s'occupe de la gestion du calendrier et visio-conférence, il traite la consultation, la planification et l'annulation de visio-conférence. La gestion de calendrier et visio-conférence nécessite une authentification d'instructeur. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification, l'instructeur peut consulter, planifier, modifier et

annuler une visio-conférence sur le calendrier. Ces relations sont réalisées par l'association extend. L'instructeur peut planifier une visio-conférence en démarrant une visio-conférence, cette relation est réalisée par l'association extend entre planifier visio-conférence et démarrer visio-conférence. Le participant peut accepter et annuler la demande de visio-conférence. Pour cela elle nécessite le démarrage d'une séance visio-conférence, ces relations sont réalisées par l'association include avec démarrer visio-conférence.

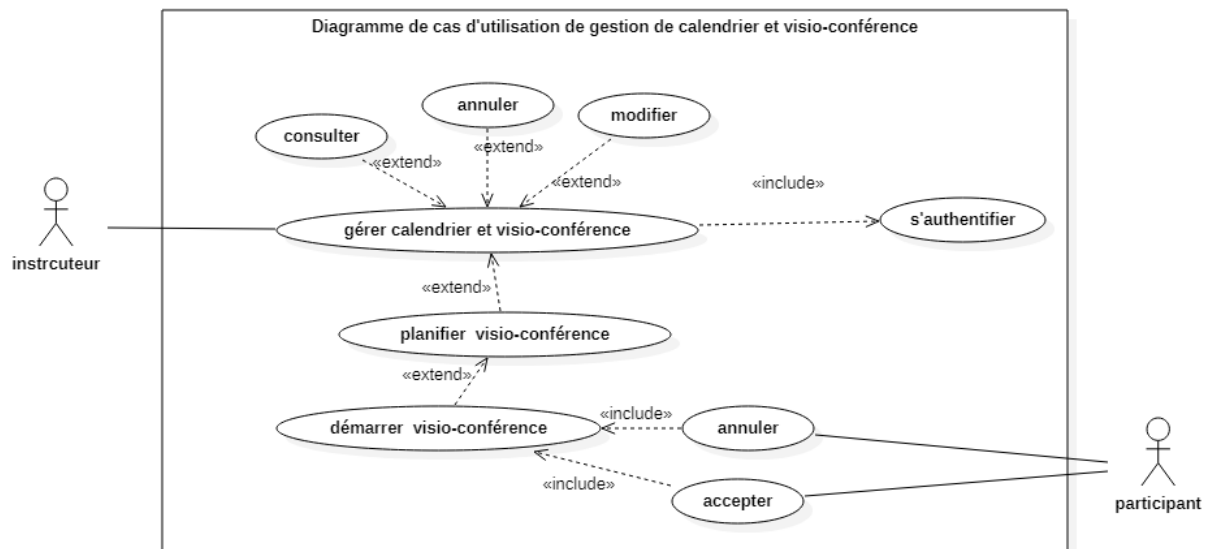


FIGURE 51:DIAGRAMME DE CAS D'UTILISATION "GESTION DE CALENDRIER ET VISIO-CONFERENCE"

1.2. Diagramme de cas d'utilisation « gérer messagerie instantanée »

Ce sprint s'occupe aussi de la gestion des messageries, il traite la sélection du/des récepteurs. La gestion de messagerie instantanée nécessite une authentification de l'utilisateur. Pour cela elle est liée par s'authentifier par une relation include. Après l'authentification, l'utilisateur peut sélectionner un récepteur. Cette relation est réalisée par l'association extend. Envoyer message et passer un appel nécessite la sélection du/des récepteurs, Ces relations sont réalisées par l'association include entre sélectionnée récepteur, envoyer message et passer appel. L'utilisateur peut envoyer un message en notifiant le récepteur, cette relation est réalisée par l'association extend entre envoyer message et notifier.

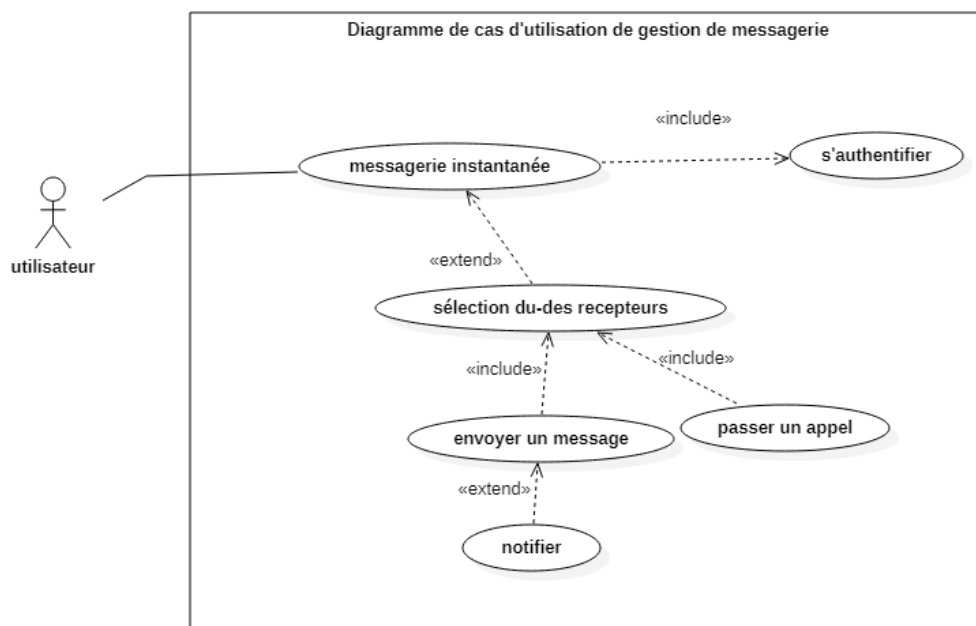


FIGURE 52:DIAGRAMME DE CAS D'UTILISATION « GESTION DE MESSAGERIE »

2. Description des cas d'utilisation

2.1. Description du cas d'utilisation « gestion de visio-conférence »

Ce tableau présentera une description du cas d'utilisation ” Planifié visio-conférence”

Titre	Planifié visio-conférence
Acteurs	Instructeur
Résumé	L'outil permet la planification de visio-conférence.
Description des enchainements	
Pré-conditions	Instructeur authentifié.
Post-condition	Visio-conférence planifiée.
Scénario nominal	
	<p>L'instructeur clique sur calendrier dans le menu principal.</p> <p>L'instructeur clique deux fois sur la date choisie.</p> <p>L'instructeur remplit les champs pour la planification.</p> <p>L'instructeur valide la planification.</p>
Scénario alternatif	
	E1 : L'instructeur saisit des données non valides ou des champs vides.

	Le système affiche un message d'erreur.
IHM	Modal avec formulaire Textview : titre Select-box : équipe Calendrier : pour le début et la fin Deux Bouton : enregistrer et annuler

TABLEAU 14:DESCRIPTION DE CAS D'UTILISATION PLANIFIER VISIO-CONFERENCE

Ce tableau présentera une description du cas d'utilisation " Modifier visio-conférence"

Titre	Modifier visio-conférence
Acteurs	Instructeur
Résumé	L'outil permet la modification de visio-conférence.
Description des enchainements	
Pré-conditions	Instructeur authentifié.
Post-condition	Visio-conférence modifiée
Scénario nominal	
	L'instructeur clique sur calendrier dans le menu principal. L'instructeur clique deux fois sur la date choisie. L'instructeur modifier le formulaire. L'instructeur enregistre la modification.
Scénario alternatif	
	E1 : L'instructeur saisit des données non valides ou des champs vides. Le système affiche un message d'erreur.
IHM	Modal avec formulaire Textview : titre Select-box : équipe Calendrier : pour le début et la fin Deux Bouton : enregistrer et annuler

TABLEAU 15 : DESCRIPTION DE CAS D'UTILISATION MODIFIER VISIO-CONFERENCE

2.2. Description du cas d'utilisation Gérer messagerie

Ce tableau présentera une description du cas d'utilisation « messagerie instantanée »

Titre	Messagerie instantanée
Acteurs	Utilisateur
Résumé	L'outil permet la messagerie instantanée.
Description des enchainements	
Pré-conditions	Utilisateur authentifié.
Post-condition	Message envoyé.
Scénario nominal	
	L'utilisateur clique sur conversation dans le menu principale. L'utilisateur choisit un récepteur. L'utilisateur écrit un message. L'utilisateur envoie un message.
Scénario alternatif	
	E1 : l'utilisateur saisit des données non valides ou des champs vides. Le système affiche un message d'erreur.

TABLEAU 16:DESCRIPTION DE CAS D'UTILISATION ENVOYER MESSAGE

IV. Conception

Dans cette partie, nous allons élucider la phase de conception, en exposant l'architecture technique et les scénarios à réaliser.

Scenarios

❖ Diagramme de séquence détaillé « gérer messagerie instantanée »

Dans cette partie, nous allons prendre en considération la partie « gestion de messagerie ». Nous allons présenter le diagramme de séquence « messagerie instantanée »

Dans ce diagramme, nous expliquerons la démarche dont un utilisateur doit suivre pour envoyer un message à la base de données des messageries, dès que l'interface s'affiche il choisit un membre, ensuite il saisit un message et il l'envoie.

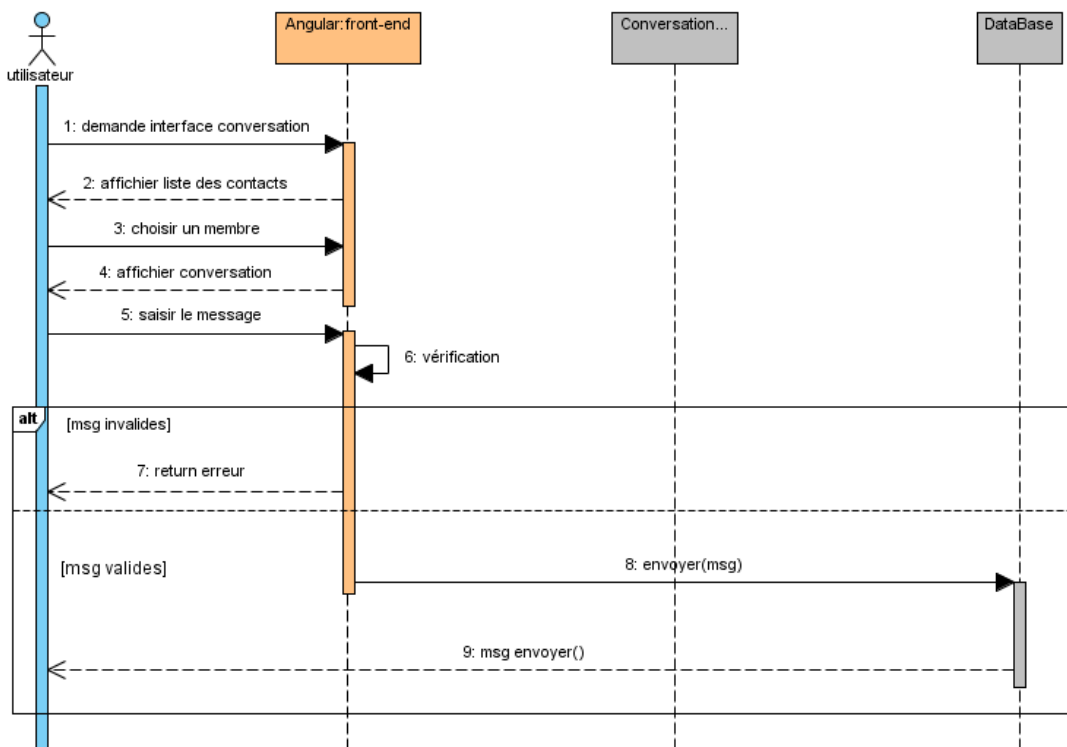


FIGURE 53:DIAGRAMME DE SEQUENCE "ENVOYER UN MESSAGE"

❖ Diagramme de séquence détaillé « envoyer notification »

Dans cette partie, nous allons prendre en considération la partie « gestion de notification ». Nous allons présenter le diagramme de séquence « envoyer notification ». Dans ce diagramme nous expliquerons la démarche dont un utilisateur doit suivre pour envoyer une notification à la base de données des notifications, dès que l'application s'ouvre il met à jour le socket puis il le publie, ensuite, si le membre est connecté, il reçoit une notification en temps réel.

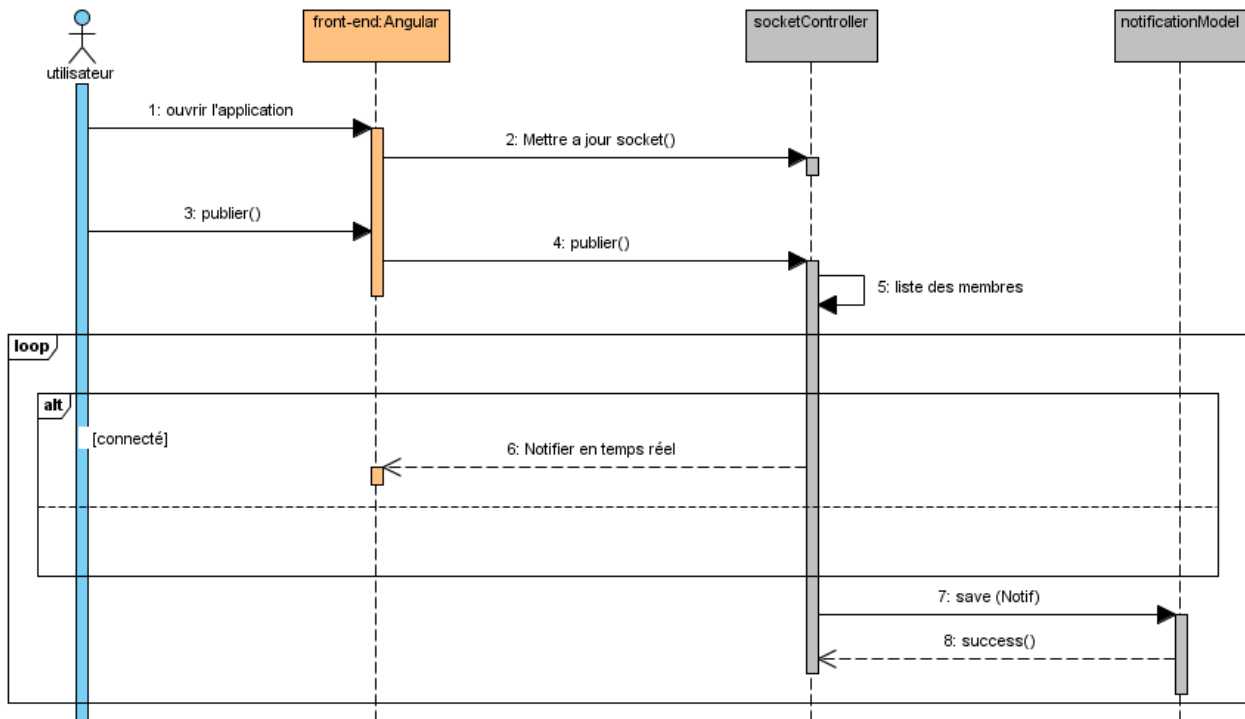


FIGURE 54:DIAGRAMME DE SEQUENCE " GESTION DE NOTIFICATION"

V. Réalisation

Cette partie présentera les différentes interfaces de la gestion des cours-online. Pour envoyer un message ou de passer un appel, l'utilisateur doit cliquer sur conversation dans le menu principale. Après, il sélectionne un membre et il saisit le message ou clique sur l'icône de l'appel.

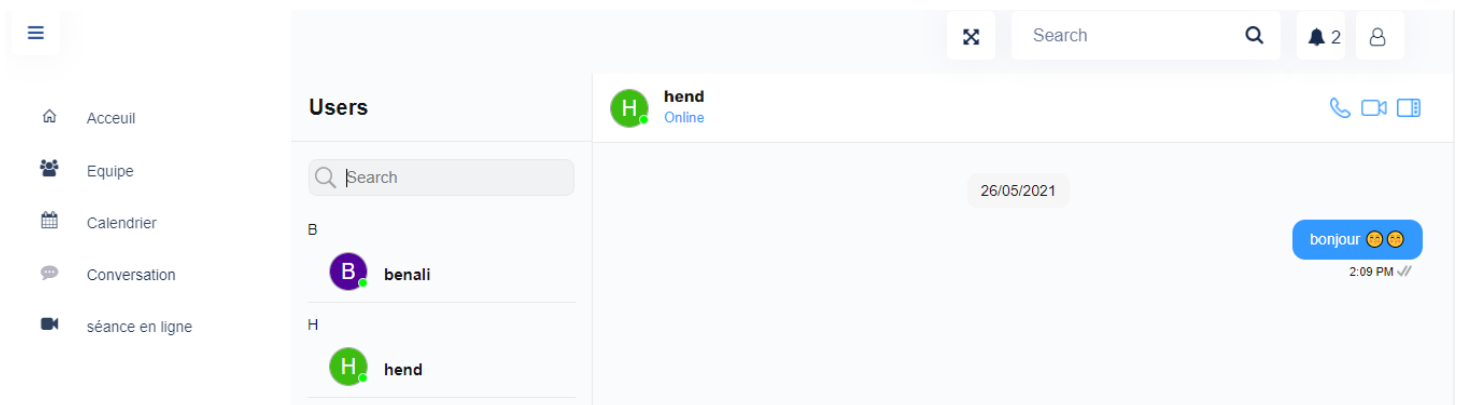


FIGURE 55:INTERFACE "GESTION DE MESSAGERIE"

Passons maintenant à la gestion des notifications. Chaque participant reçoit une notification lors de la publication d'un test. Le cadre en rouge représente la notification.

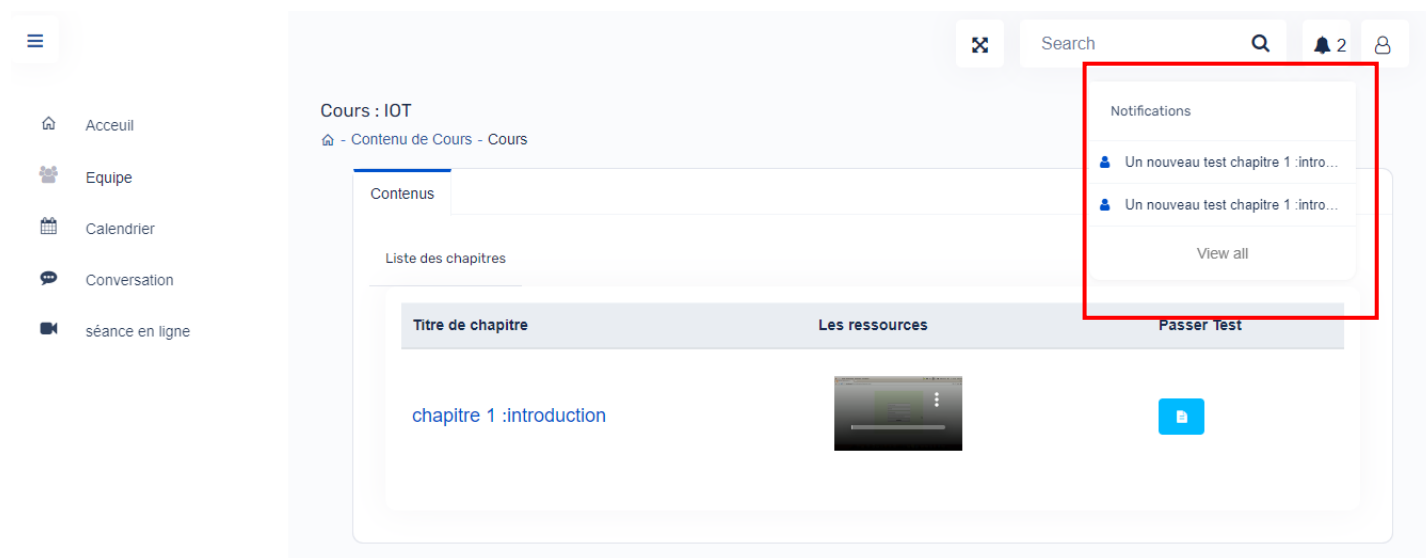


FIGURE 56:INTERFACE "GESTION DE NOTIFICATION"

Passons maintenant à la gestion du calendrier ou bien de visio-conférence. Cette figure représente l'interface du calendrier.

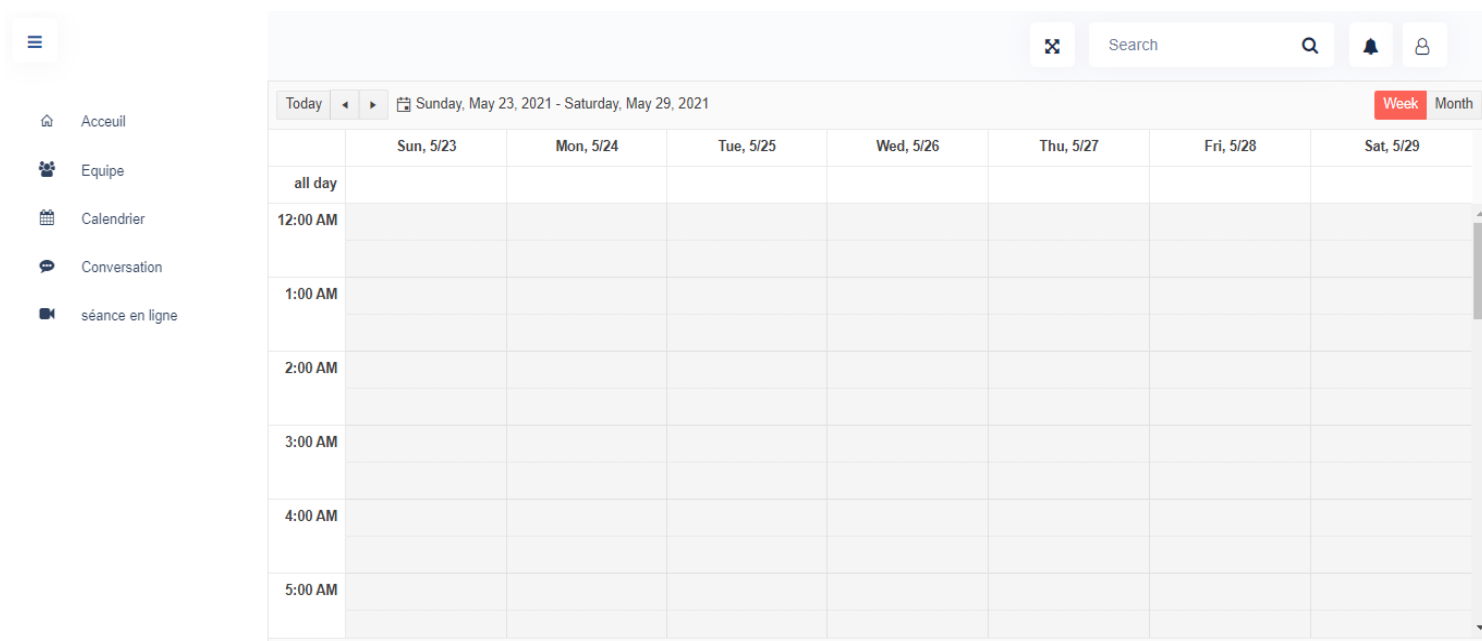


FIGURE 57:INTERFACE " GESTION DE CALENDRIER"

Pour planifier une visio-conférence, l'instructeur clique deux fois sur la date qu'il choisit. Une modal s'affiche. il doit remplir les champs et valide. La figure suivante représente l'interface de la planification d'une visio-conférence.

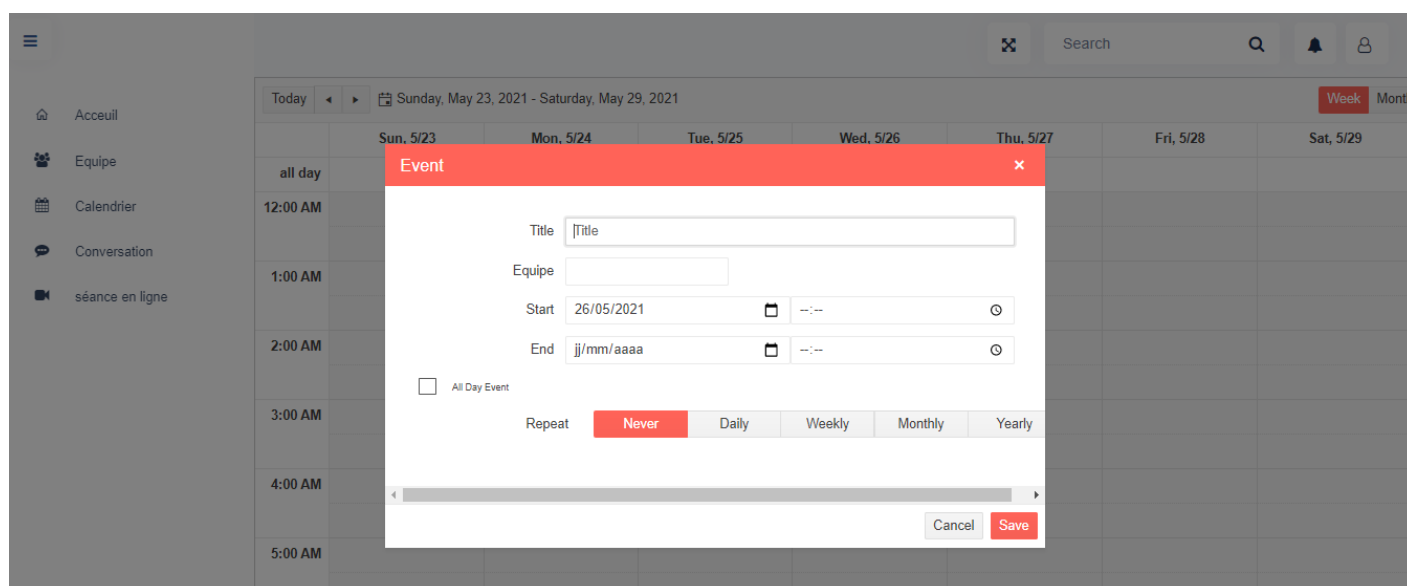


FIGURE 58:INTERFACE "PLANIFIER VISIO-CONFERENCE"

Pour modifier quelque chose pour la visio-conférence, l'instructeur clique deux fois sur l'évènement. Le modal s'affiche avec les données. La figure suivante représente l'interface de la modification d'une visio-conférence.

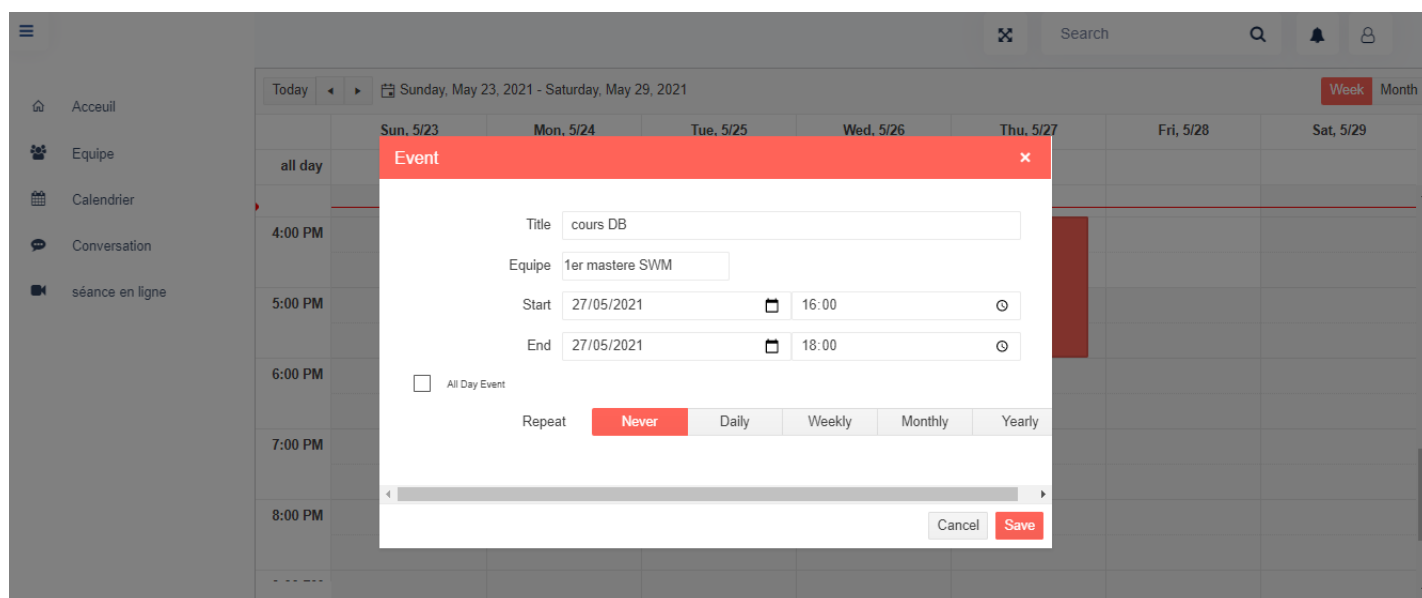


FIGURE 59:INTERFACE " MODIFIER VISIO-CONFERENCE"

En cliquant sur l'icône✕, supprime une visio-conférence, un message d'avertissement présenté dans la figure suivant s'affiche.

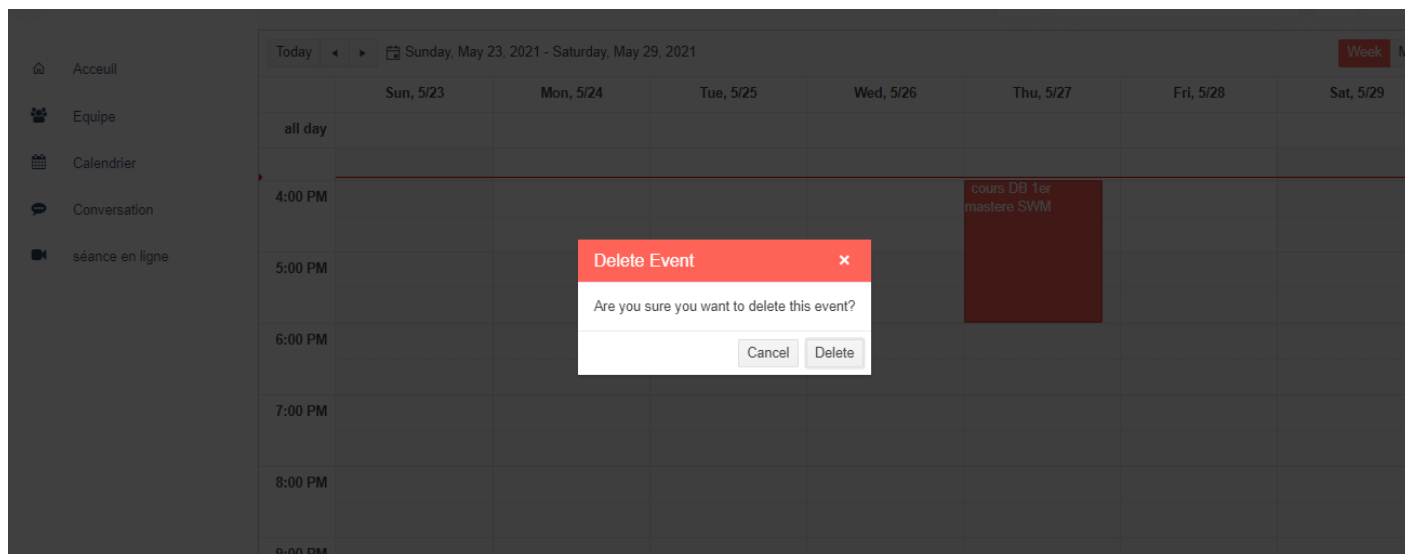


FIGURE 60:INTERFACE " SUPPRIMER VISIO-CONFERENCE"

L'instructeur choisi une équipe et il lance le cours online. Cette figure représente l'interface de démarrage de cours online ou bien la visio-conférence.

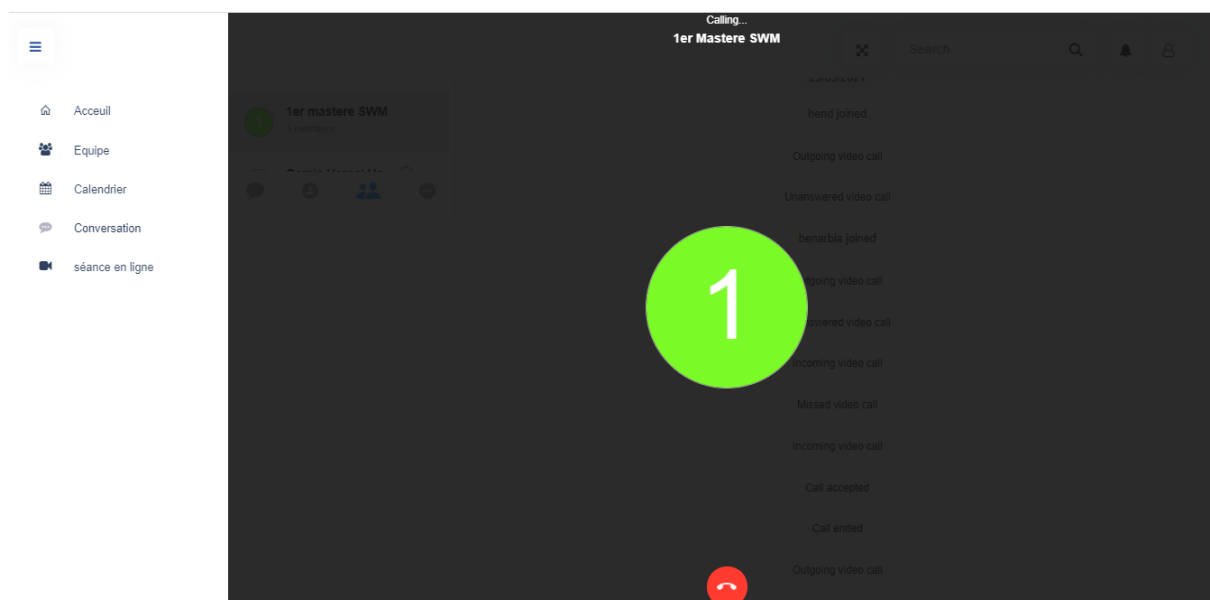


FIGURE 61:INTERFACE "DEMARRER UNE VISIO-CONFERENCE "

Après le démarrage de cours, chaque membre de l'équipe peut accepter ou annuler la participation à ce cours, comme illustré sur la figure suivante.

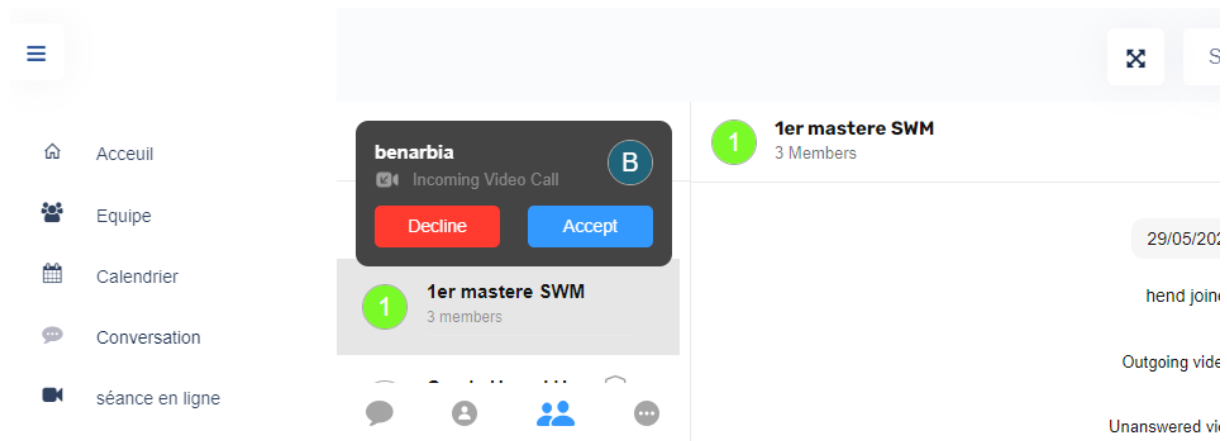


FIGURE 62:INTERFACE "ACCEPTER OU REFUSER LA PARTICIPATION"

La figure suivante représente l'interface Lorsque le participant accepter la demande de la participation au cours.

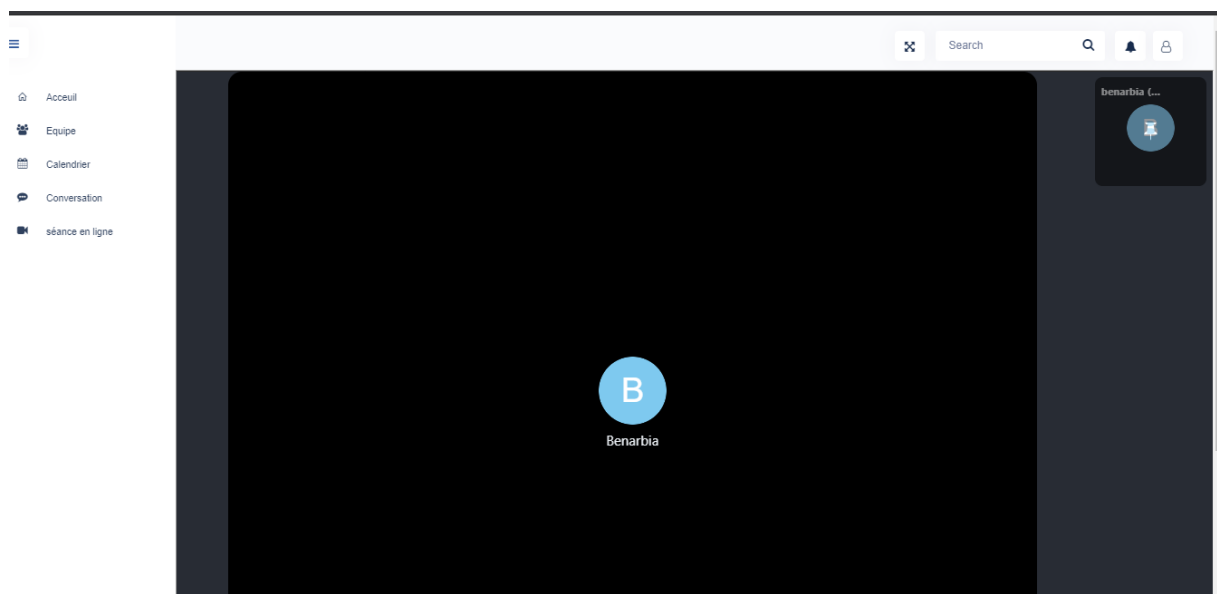


FIGURE 63:INTERFACE DE PARTICIPANT

Cette figure représente l'interface de l'instructeur, il permet de partager l'écran, d'ouvrir la caméra et l'audio

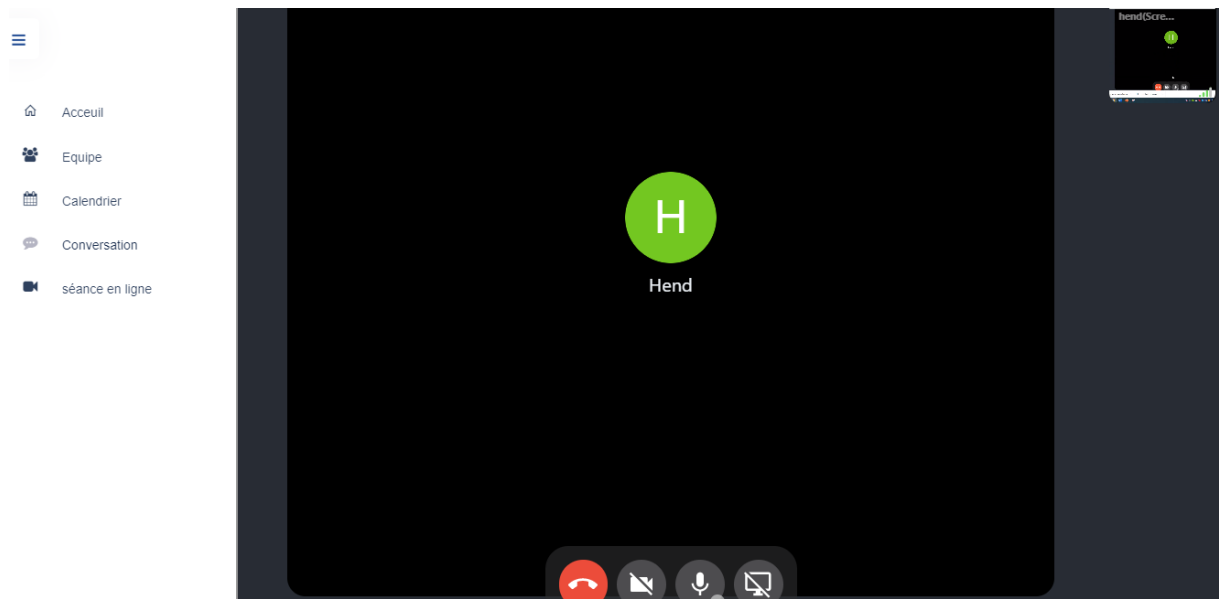


FIGURE 64:INTERFACE DE L'INSTRUCTEUR

VI. Conclusion

Nous avons réussi à développer le dernier sprint de notre Cross-Plateforme. Notre application web et mobile est prête à être exploitée en offrant aux utilisateurs la possibilité de gérer convenablement toutes les fonctionnalités de Cross-Plateforme.

Conclusion générale

Ce stage de fin de parcours a été très enrichissant, car il nous a permis de découvrir les détails de métier du développeur, aussi il nous a permis de participer concrètement à ses enjeux au travers de nos missions variées que nous avons particulièrement appréciées.

La société DEV-GX qui nous a accueilli pendant 4 mois nous a donné la chance de suivre une méthodologie de travail bien étudiée et de manipuler des techniques innovantes et évolutives.

Nous avons été sollicités en tant que stagiaires pour réaliser un projet de fin d'études pour l'obtention du diplôme national de Mastère Professionnel en Service Web et Multimédia à l'institut supérieur d'informatique et des techniques de communication de Hammam Sousse. L'objectif de ce projet de fin d'études était de créer une Cross-Plateforme pour l'enseignement en ligne bien déterminé.

Contrairement à la majorité des applications existantes sur le marché qui possèdent plusieurs inconvénients comme le temps limité de visio-conférence, le nombre des participants limité et existe deux modes d'enseignement chacun ses fonctionnalités. Nous avons réalisé un système qui permet à la fois de gérer l'enseignement online et l'enseignement offline.

Pendant le déroulement de notre stage, nous avons eu l'opportunité de travailler sur plusieurs aspects. L'effort réalisé s'est avéré très valorisant pour notre expérience professionnelle aussi bien en ce qui concerne le domaine technique que le relationnel et le développement personnel.

Nous avons pu acquérir des compétences supplémentaires en développement et surtout en gestion de temps.

On a utilisé la méthode Scrum qui est basé sur des sprints, dans le cas de ce rapport, nous avons effectué le travail demandé en trois sprints. La conception Scrum nous a permis de concevoir avec succès chaque sprint pour le tester et l'implémenter séparément.

Nous avons bien développé des connaissances techniques sur plusieurs outils : Angular11 pour le front-end, NodeJs, ExpressJs pour le backend, l'architecture MVC, le système de gestion de base de données MongoDB, Cordova pour l'environnement mobile et de maîtriser le langage de modélisation UML.

La collaboration au sein de l'équipe a participé dans la réussite de notre travail vu que tous les postes sont liés et doivent communiquer entre eux. Une bonne ambiance règne dans DEV-GX et tout le personnel a été très coopératif et attentif à nos questions.

Et après notre rapide intégration dans l'équipe, nous avons eu l'occasion de réaliser plusieurs tâches qui ont constitué une mission de stage globale.

Ce stage a parfaitement répondu à nos attentes car nous souhaitons découvrir les nouvelles missions, il nous a permis de découvrir un univers que nous ne connaissons finalement que très peu mais pour lequel nous portons un immense intérêt.

Faute de temps, nous n'avons pas pu terminer le travail. Alors, comme perspective, il y a plusieurs améliorations et fonctionnalités qu'on peut ajouter à notre application notamment l'ajout de l'enregistrement de cours online et développement d'une application desktop.

Bibliographie

- [1] <https://www.01net.com/telecharger/windows/Programmation/creation/fiches/130819.html>
- [2] <https://www.mongodb.com/fr>
- [3] <https://monpetitdev.fr/cest-quoi-angular-definition/>
- [4] <https://grafikart.fr/tutoriels/nodejs-intro-792>
- [5] <https://actualiteinformatique.fr/data/definition-nosql-not-only-sql-database>
- [6] <https://fr.wikipedia.org/wiki/Express.js>
- [7] https://fr.wikipedia.org/wiki/Android_Studio
- [8] <http://www.lsis.org/elmouelhia/courses/php/coursMVC.pdf>
- [9] <http://www.oitsfax.org/files/PFE%20ISIMS%202012.pdf>
- [10] <https://www.capterra.fr/software/186631/google-classroom>
- [11] <https://www.tableauxinteractifs.fr/logiciels-tni/comparatifs-de-logiciels-tni/comparatif-de-logiciels-de-visioconference-en-enseignement-zoom-vs-microsoft-teams>