

Team: iDev

Membres: Anas Houissa, Mohamed Aziz Ben Hmida, Malek Zommit Chatti, Isslem Ouederni, Oussema Dengel, Meriam Ben Ida

[PI selon 2TUP](#)

Branche fonctionnelle	
Capture des besoins	<p>Cas d'utilisation 1: Ajouter une publication dans la rubrique club</p> <p>Préconditions:</p> <ul style="list-style-type: none">· l'utilisateur doit être authentifié· l'utilisateur doit être le responsable de ce club <p>Enchainements:</p> <ul style="list-style-type: none">· Se connecter· Consulter la rubrique de son club· Rédiger le texte de la publication· Ajouter un fichier/une image· Cliquer sur le bouton publier <p>Exceptions:</p> <ul style="list-style-type: none">· Le texte d'une publication ne peut pas être vide. <p>Postconditions:</p> <ul style="list-style-type: none">· Message indiquant la publication est en attente d'approbation par l'admin est affiché.· L'interface de la rubrique est encore affichée et le responsable peut ajouter une autre publication. <p>Contraintes non-fonctionnelles</p> <p>Confidentialité:</p> <ul style="list-style-type: none">· L'utilisateur doit être authentifié. <p>Fréquence de cas d'utilisation:</p> <ul style="list-style-type: none">· plusieurs fois par jour <p>Disponibilité:</p> <ul style="list-style-type: none">· ce cas d'utilisation est toujours disponible. <p>Concurrence:</p> <ul style="list-style-type: none">· Seulement les responsables des clubs peuvent réaliser ce cas d'utilisation. <p>Intégrité:</p> <ul style="list-style-type: none">· Il peut filtrer les autres publications par date· Il peut supprimer/modifier une publication <p>Volumétrie:</p> <ul style="list-style-type: none">· Un seul acteur, le responsable d'un club. <p>*****</p> <p>*****</p>

	<p>Cas d'utilisation 2: Publier un document dans le centre de partage</p> <p>Pré-conditions: L'étudiant doit être authentifié et son compte ne doit pas être banni</p> <p>Enchaînements: L'étudiant choisit un nom, sélectionne un niveau et une matière L'étudiant attache un fichier ou saisit une url valide L'étudiant clique sur le bouton "ajouter" Un message de succès est affiché</p> <p>Exceptions: L'extension du fichier attaché n'est pas autorisées L'url saisie est invalide Les champs ne sont pas tous remplis</p> <p>Post-conditions: Un document est créé dans le centre de partage</p> <p>Contraintes non-fonctionnelles:</p> <p>Confidentialité: L'étudiant doit avoir le droit d'accès à la plateforme.</p> <p>Fréquence de cas d'utilisation: séquentiellement</p> <p>Disponibilité: toujours</p> <p>Volumétrie: un seul acteur, un seul message à la fois</p> <p>*****</p> <p>*****</p> <p>Cas d'utilisation 3: Modifier une publication de tableau de bord</p> <p>Pré-conditions: L'administrateur doit être authentifié sur la plateforme Edspace</p> <p>Enchaînement :</p> <ul style="list-style-type: none"> ● L'administrateur choisit la publication à modifier . ● L'administrateur remplit les champs à modifier . ● L'administrateur appuie sur le bouton "Modifier". <p>Exceptions :</p> <ul style="list-style-type: none"> ● Un ou plusieurs champs ne sont pas remplis ● L'extension de l'image n'est pas jpeg, png ou jpg
--	--

Post-conditions: La publication est modifiée.

Contraintes non-fonctionnelles :

Confidentialité: L'administrateur a un compte Edspace

Fréquence de cas d'utilisation : plusieurs fois par jour.

Disponibilité: Durants les jours de travail

Intégrité :

- Il peut modifier ajouter une publication
- Il peut supprimer une publication

Volumétrie : Les différents agents de scolarité d'un établissement

cas d'utilisation 4: Envoyer d'un message

pr2-conditions :

- l'utilisateur doit être authentifié
- l'utilisateur doit être un étudiant

enchaînements:

- Se connecter
- est sous le chatbox
- écrit son message
- cliquer sur le bouton "send"

exceptions:

- l'étudiant doit envoi un message non-vide

Postconditions:

- Le message est envoyé et affiché dans le chatbox.

Contraintes non-fonctionnelles:

Confidentialité:

- L'étudiant doit être authentifié.

Fréquence de cas d'utilisation:

- Plusieurs fois par jour.

Disponibilité:

- Toujours.

Intégrité:

- Il peut chercher des messages.

Volumétrie :

- Un seul étudiant.

Cas d'utilisation 5 :répondre à une question

Préconditions : il faut que la question ne soit pas vérifiées

Enchainements : premièrement il faut que l'utilisateur soit connecté, puis il faut accéder au partie forum, ensuite il faut choisir la question à répondre et enfin écrire la réponse et valider.

Exceptions : si le champ est vide il y aura un message d'erreur.

Postconditions: au niveau du système, il y aura appel au service ReponseService, la méthode add Réponse spécifiquement, et celle là réagit avec la base de données, si la requête http est bien rédigé il y aura un retour d'une réponse http 201 sinon il y aura un retour d'une réponse 500 "bad request"

Contraintes non-fonctionnelles

Confidentialité: accès au système.

Fréquence de cas d'utilisation: plusieurs fois par jour

Disponibilité: tout jour disponible

Concurrence: Student et administrateur

Intégrité: Poser une question et repondre a des questions.

Volumétrie: Tous les acteurs.

Cas d'utilisation 6: Authentification

	<p>Préconditions: l'utilisateur doit avoir déjà un compte</p> <p>Enchaînements: l'utilisateur remplit les champs avec son email et son mot de passe après pour l'action de cette cas d'utilisation se fait il faut cliquer sur le bouton login</p> <p>Exceptions: si l'email ou le mot de passe saisie est incorrecte donc l'authentification ne fonctionne pas ou bien les champs sont vides</p> <p>Postconditions: l'utilisateur s'authentifier avec succès et la rubrique du tableau d'affichage s'affiche</p> <p style="text-align: center;"><i>Contraintes non-fonctionnelles</i></p> <p>Confidentialité: accès au système, l'utilisateur ne doit pas être banni</p> <p>Fréquence de cas d'utilisation: plusieurs fois par jour</p> <p>Disponibilité: ce cas d'utilisation est disponible au cas ou l'utilisateur n'est pas banni</p> <p>Concurrence: tous les utilisateurs qu'ils ne sont pas bannis peuvent réaliser ce cas d'utilisation</p> <p>Intégrité: si l'utilisateur oublie il peut le récupérer vers un email de récupération .</p> <p>Volumétrie: tous les acteurs</p>
Analyse	<ul style="list-style-type: none"> ● <i>Regroupement en catégories</i> GUIs: Les interfaces graphiques Services: Les contrôleurs Entités: Les modèles des objets Utils: Les Singletons, Les variables statiques Tests: Application Main de test ● <i>Développement du modèle statique: Objets et attributs</i> User: int id , String username , String prenom , String email , String password , Boolean isBanned

	<p>Thread : int id, string question, int user_id, date threadDate, boolean display.</p> <p>ThreadType : int id, String content, boolean display</p> <p>Reponse : int id, String reply, Date replyDate, int user_id, boolean display, int thread_id.</p> <p>Club:Clubdesc:String</p> <p>Club'Clubid:int,</p> <p>ClubPub;ClubPubDesc: String</p> <p>Document: id int, niveau Niveau (niveau existant dans la bd), matiere Matiere (matière existante dans la bd), url String (url valide), fichier String (extension autorisée)</p> <p>Matiere: id String, niveau Niveau (niveau existant dans la bd)</p> <p>Classe:id int,classe String, niveau Niveau</p> <p>Niveau:id String</p> <p>Message:id int, content String, postDate Date, user User , classe Classe</p> <p>PublicationNews: id int, title String, owner String, content String, image String, CategoryName CategoryNews</p> <p>Emploi: id int, title String, content String, image String, CategoryName CategoryEmploi</p> <p>CategoryEmploi: id int, categoryName String</p> <p>CategoryNews: id int, categoryName String</p> <p>● Réalisation des cas d'utilisation</p> <p>Clubs universitaires:</p> <ul style="list-style-type: none"> - afficher les clubs - modifier les clubs - supprimer les clubs - ajouter des clubs
--	--

Rubriques des clubs:

- afficher les publications
- modifier des publications
- supprimer des publications
- ajouter des publications

Gestion d'utilisateur:

- afficher la liste des administrateurs
- modifier un administrateur
- ajouter un administrateur
- Supprimer un administrateur
- afficher la liste des étudiants
- ajouter des etudiants
- modifier un étudiant
- supprimer un étudiant

Centre de partage:

- ajouter un document
- afficher les documents
- modifier un document
- supprimer un document

Matière:

- ajouter une matière
- afficher les matières
- modifier une matière
- supprimer une matière

Thread:

- Ajouter une question
- Modifier une question
- Supprimer une question
- Afficher les questions
- Afficher une question par id
- Chercher la question sur google

Réponse:

- Ajouter une réponse

	<ul style="list-style-type: none"> - Modifier une réponse - Supprimer une réponse - Afficher toutes les réponses - Afficher les réponses par question. <p>ThreadType:</p> <ul style="list-style-type: none"> - Ajouter un topic - Modifier un topic - Supprimer un topic - Afficher les topics - Afficher les questions par topic <p>Niveau:</p> <ul style="list-style-type: none"> - Ajouter un niveau - afficher les niveaux - modifier un niveau - supprimer un niveau <p>Classe:</p> <ul style="list-style-type: none"> - ajouter une classe - afficher les classes - modifier une classe - supprimer une classe <p>Message:</p> <ul style="list-style-type: none"> - ajouter un message - afficher les messages - modifier un message - supprimer un message <p>Publication News:</p> <ul style="list-style-type: none"> - Ajouter une publication - modifier une publication - supprimer une publication - afficher la liste des publications - afficher une publication <p>Emploi :</p>
--	--

- Ajouter une offre d'emploi
- modifier une offre d'emploi
- supprimer une offre d'emploi
- afficher la liste des offres d'emploi
- afficher une offre d'emploi

Catégorie News :

- Ajouter une catégorie
- modifier une catégorie
- supprimer une catégorie
- afficher la liste des catégories

Catégorie Emplois :

- Ajouter une catégorie
- modifier une catégorie
- supprimer une catégorie
- afficher la liste des catégories

- ***Création des classes frontières (sert à récupérer et à insérer des données dans la base), contrôleurs (fait l'articulation entre la frontière et les classes métiers) et collections d'objets (représenter un ensemble d'objets)***

Centre de partage:

- DocumentService: classe frontière
- DocumentsList: contrôleur
- Document: entité

Matière:

- MatiereService: classe frontière
- MatiereList: contrôleur
- Matière: entité

Publication News:

- NewsService : Classe frontiere
- AllNews : Controlleur
- AddNews : Controlleur

- UpdateNews: Controlleur
- News : Entité

Emploi :

- EmploiService : Classe frontière
- AllEmploi : Contrôleur
- AddEmploi : Contrôleur
- UpdateEmploi: Controlleur
- Emploi : Entité

Catégorie News :

- NewsCategoryService : Classe frontière
- AllCategories : Contrôleur
- AddCategory : Contrôleur
- UpdateCategory: Controlleur
- CategoryNews : Entité

Catégorie Emploi :

- EmploiCategoryService : Classe frontière
- AllCategories : Contrôleur
- AddCategory : Contrôleur
- UpdateCategory: Contrôleur
- CategoryEmploi : Entité

Clubs universitaires:

- ClubService
- ClubList
- Club

Rubriques des clubs:

- ClubPubService
- ClubPubsList
- ClubPub

Niveau:

- NiveauService : Frontière
- Niveau: Entité
- NiveauList: Controller

Classe:

- ClasseService: Frontière
- Classe: Entité
- ClasseList: Controller

Message:

- MessageService: Frontière
- Message: Entité
- MessageList: Controller

Utilisateurs:

- UserService : classe frontière
- Login : Contrôleur
- StudentService : classe frontière
- StudentList : Contrôleur
- AddStudent : Contrôleur
- UpdateStudent: Contrôleur
- AdminService :classe frontière
- AdminList :Contrôleur
- AddAdmin :Contrôleur
- UpdateAdmin :Contrôleur
- User : entité

Thread :

- ThreadService : classe frontière.
- Thread : entité.
- AddThread : contrôleur.

Reponse :

- ReponseService : classe frontière.
- Reponse : entité .
- AddReponse : contrôleur.

ThreadType :

- TopicService : classe frontière.
- ThreadType: entité.
- AddTopic: contrôleur.

Branche technique	
Capture des besoins techniques	Environnement matériel 6 PCs d'un minimum de 8GB de mémoire RAM Système d'exploitation : Windows 10 Environnement logiciel Gestion du projet: Trello, github, LucidChart Outils de développement : NetBeans SceneBuilder Technologie utilisée : JavaFX, MySql
Conception générale	Composants nécessaires à la construction de l'architecture technique: Serveur d'application : Apache Serveur de base de donnée : MySQL (artifact: script.sql) Pc (artifact: exécutable de l'application Desktop)

Phase de réalisation	
Itération 1	
Conception préliminaire	Création du diagramme de déploiement
Conception générale	La définitions des différentes interfaces de l'application, qui vont être développés
Conception détaillée	Conception des: <ul style="list-style-type: none"> - classes et des attributs - classes d'associations - des méthodes - modèle relationnel
Codage et test	<ul style="list-style-type: none"> - Création des différents paquets selon le modèle MVC. - Développement des méthodes du CRUD des différents modules de l'application. - Développement d'une interface graphique pour chaque fonction de base de chaque module. - Test de chaque fonction et correction des erreurs avec quelques modifications pour améliorer l'ergonomie de l'interface.
Recette	Phase de validation

