

Projet Fédérateur

Professional social media for universities web application

NAJJAR Younes

KHADIM Safaa

IDATMANE Zakaria

EL AITAR Anass

Sous la direction :

Pr. EL HASSOUNY Azeddine

Résumé

Dans le cadre du projet fédérateur nous avons choisi le sujet “Professional Social media for universities” une idée que nous voulions réaliser par la suite au sein de notre école l’ ENSIAS et par la suite dans les différents universités du monde pour cela nous avons réalisé dans un premier temps une étude de besoins afin d’avoir un référentiel d’exigence et puis une vision stratégique. Dans une deuxième partie d’analyse nous sommes basées sur des articles de recherches pour étudier la psychologie des réseaux sociaux afin de mettre en valeur quelques éléments dans notre application, ensuite nous avons défini l’architecture utilisée pour le moment et celle qu’on doit utiliser ultérieurement. Dans la troisième partie nous avons conçu notre application. Et finalement dans la quatrième partie c’est la réalisation.

Table de matière

Résumé	2
Table de matière	3
Présentation du projet	6
1- Le but de notre projet	7
2- Naissance d'idée	7
3- Cible :	8
4- l'identification des besoins	8
4-1 statistiques au niveau internationale	9
4-2 statistiques aux niveaux de l'ENSIAS	9
les élèves	9
les professeurs:	13
4-3 Conclusion	14
5- Référentiel des exigences	16
5-1 Exigences fonctionnelles :	16
5-2 Exigences non fonctionnelles	17
6- Vision stratégique :	17
6-1 Frais :	18
6-2 Revenus :	20
6-3 Vision stratégique pour 5 ans :	21
Analyse	22
1-étude psychologique:	23
1-1 Comment fonctionne la psychologie des réseaux sociaux	23
1-1-1 la notion de récompense	23
1-1-2 le rôle du système de partage	23
1-2 Discourse analysis	24
1-3 conclusion:	26
2- UX Design	26
L'utilisabilité :	27
Le hook canevas:	27
adapter le design thinking:	29
3- Architecture	30

3-1 architecture monolithique	30
3-2 microservices architecture	32
3-3 monolithique vs microservices	34
3-4 Décision:	34
4- architecture monolithique de l'application:	36
4-1 Architecture 2 tiers	36
4-2 Architecture modulaire	36
5- architecture en microservices prochainement:	39
5-1 Exemple des entreprises utilisant architecture Microservices:	39
5-2 les patterns de la décomposition	40
5-3 les microservices et l'apprentissage automatique	41
5-4 microservice ecosystem	42
5-5 la démarche de la décomposition choisi pour notre application	43
5-6 application de la démarche : La décomposition de notre application	46
5-6-1 les capacités métiers:	46
5-6-2 les microservices de notre application et les langages de développement et les bases de données	47
Conception	49
1- diagramme use case	50
2- diagramme de séquence	51
2-1 création de compte de l'université	51
2-2 génération des comptes des étudiants	52
3- diagramme entité relation	53
Réalisation	54
1-outils	55
1-1-backend	55
1-1 nodejs	55
1-2 express	55
1-2 frontend	56
1-3 base de données	57
1-postgresql:	57
2-hebergement de la base de donnée	57
3- problèmes rencontrées lors de l'hébergement	58
1-4 quelque librairies utilisées	59
1-sendgrid	59
2- nodemailer-sendgrid-transport	61
3- nodemailer	61
4- multer	61
Multer est un middleware node.js pour gérer les données en plusieurs parties / formulaires, qui est principalement utilisé pour télécharger des fichiers.	61
5- xlsx	61

6- morgan	61
2- Architecture du code	62
2-1 models	63
2-2 controllers	63
2-3 helpers	64
3-Screenshots	65
Bibliographie	71

Présentation du projet

Dans cette partie nous allons indiquer le but du projet ainsi que la naissance de l'idée et puis l'identification des besoins à l'aide de différents statistiques après avoir réaliser des interview avec des professeur et partager des questionnaires avec les élèves. Finalement nous allons présenter une vision stratégique du projet

1- Le but de notre projet

Notre application « professional social media for universities » est un réseau social à usage exclusivement professionnel. Alors il se distingue des autres réseaux sociaux existant par le fait qu'il est entièrement voué à des fonctions professionnelles et ayant pour but d'améliorer la connectivité et l'échange entre les universités du monde. Ce réseau permet de regrouper les universités du monde donc chaque université crée son propre compte lui permettant de créer des groupes pour ses étudiants ainsi que le corps professoral et administratif. L'application va fournir un espace privée pour chaque université permettant de la définir comme entité qui peut établir des relations avec d'autres universités

2- Naissance d'idée

A l'ENSIAS l'échange des informations entre les élèves et l'administration se fait à l'aide d'un groupe Facebook « ENSIAS Scolarité ». Nous constatons que la plupart des universités disposent d'un groupe pareil. Or ce groupe a plusieurs défauts, d'une part ce groupe ne fournit pas des informations sur les élèves comme l'année d'étude, la filière, le type du diplôme (master, cycle...) donc les publications destinées à un groupe spécifique sont partagées avec tout le monde, la création de multiple groupe pour le partage sur Facebook ne sera pas pratique. D'une autre part les professeurs ne font pas partie des groupes donc ne peuvent pas interagir avec les étudiants ni poster dans le groupe les devoirs ... Vu tout cela nous avons eu l'idée de créer un réseau social professionnel permettant de créer un espace privée pour chaque université et

ainsi interagir avec des autres universités tout en partageant les événements avec eux comme l'événement des olympiades, japan day

3- Cible :

Tous les étudiants et le corps professoral et administratif des universités Voici les types des utilisateurs :

- Administrateur général : c'est l'administrateur de toute l'application il peut accepter ou décliner les demandes des universités qui souhaitent créer leur compte
- Administrateur de l'école : ce dernier envoie une demande à l'administrateur général pour créer son compte, une fois créé il ajoute les informations de tous les étudiants et le corps professoral et administratif de l'université, à l'aide de ses informations les comptes de ses derniers sont générés automatiquement les utilisateurs concernées seront alors notifiés.
- Étudiants
- Professeurs

4- l'identification des besoins

Dans le monde compétitif d'aujourd'hui, la satisfaction des clients est la clé de la réussite. Car un client perdu, c'est d'abord une perte d'image pour l'entreprise. On dit qu'un client mécontent en parle à dix personnes, un client satisfait à trois.

L'insatisfaction du client découle le plus souvent d'une mauvaise compréhension de ses besoins par l'entreprise. Pour cela il faut bien identifier les besoins :

En se basant sur les statistiques suivantes :

4-1 statistiques au niveau internationale

1-Près de 80% des étudiants utilisent Facebook pour travailler en groupe [1] :

Le site Diplomeo a interrogé les jeunes de 17 à 27 ans sur leurs pratiques digitales. Cette enquête indique que le réseau social Facebook est désormais un outil de révision. Outre la possibilité de rester au courant des dernières actualités (42 %) et de parler avec leurs proches (40 %), les étudiants utilisent l'invention de Mark Zuckerberg pour travailler. Ils sont près de 8 sur 10 à avoir déjà «utilisé un groupe Facebook pour travailler avec leurs camarades». D'ailleurs, 59 % d'entre eux utilisent ce réseau social au moins une fois par semaine pour travailler avec les autres élèves de leur promotion.

2-les étudiants perdent actuellement leur temps sur facebook en autre choses que les études [2]

4-2 statistiques aux niveaux de l'ENSIAS

1. les élèves

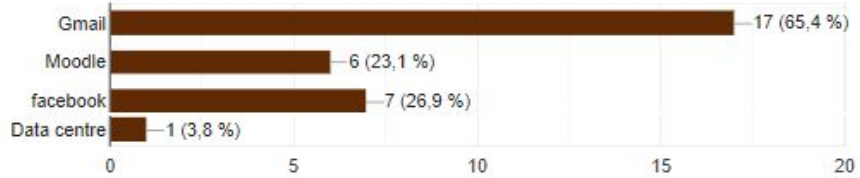
nous avons réalisé un questionnaire et nous l'avons partagé sur facebook pour savoir l'opinion des élèves de l'ENSIAS concernant la procédure du partage des documents et communication avec les profs ainsi que le suivi des projets académiques

- Partage des documents et communication avec les profs

Partage des documents et communication avec les profs

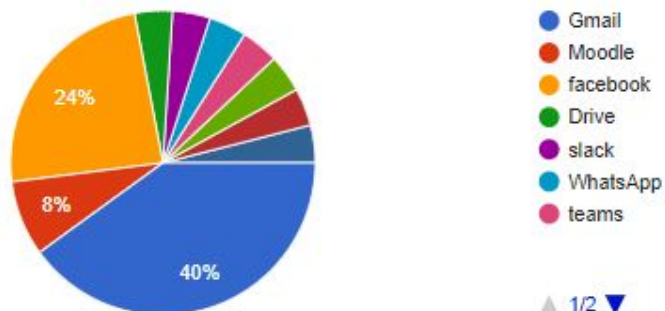
Comment les profs partagent avec vous les documents (tps, cours...) ?

26 réponses



Quelle est votre plateforme de partage préférée ?

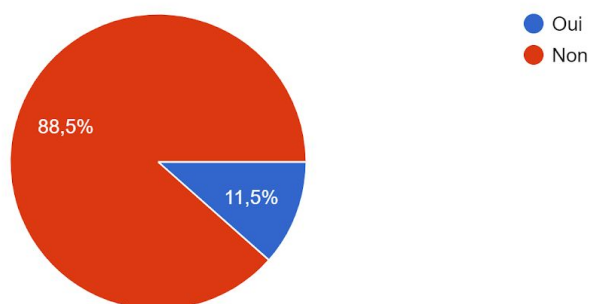
25 réponses



▲ 1/2 ▼

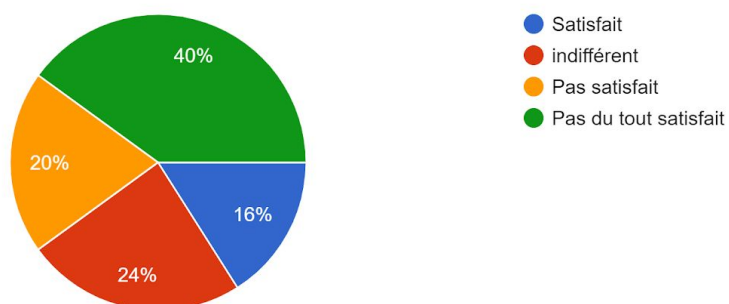
Est-ce que vous recevez les documents à temps ?

26 réponses



Est-ce que vous êtes satisfaits par la manière de partage ?

25 réponses



Quelles sont les problèmes de communication avec les profs ?

10 réponses

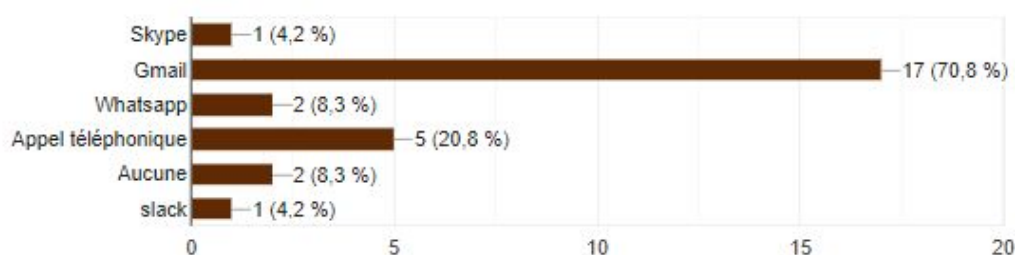
il faut unifier les methodes de communication, en fait c'est pas sorcier, l'administration peut facilement interragir et unifier la methode de partage en une seule qui satisfera la plupart des parties.

La boîte G-mail lorsqu'elle est est pleine, ça sera difficile de vérifier tous les emailspour en apercevoir qu'il y en a qui concerne les études. Il est préférable qu'il existe une plateforme dédié spécialement aux études et qui offrent plus de services que de simple transfert de fichier.

- **Projet académique**

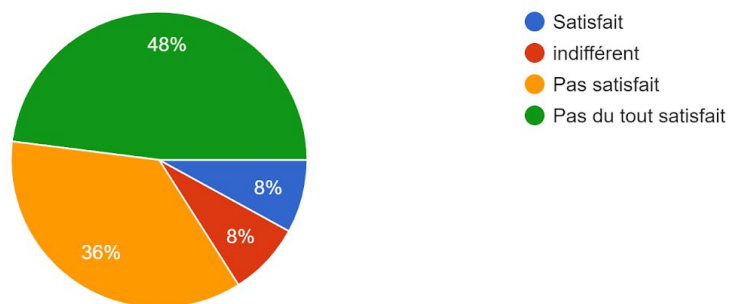
Quelles sont les plateformes que vous utilisez pour établir la communication avec vos encadrants et faire le suivi du travail ?

24 réponses



Est-ce que vous êtes satisfaits par la procédure du suivi des projets et d'encadrement ?

25 réponses



Quels sont les problèmes que vous rencontrez le plus ?

10 réponses

Le retard de la réponse

mon université ensias n'a pas de plateforme ou d'espace de travail spécial, toutes les communications se font via Facebook ou Gmail et aucune d'entre elles n'est satisfaisante

Écrire un mail parfois n'est pas suffisant par exemple pour citer des problèmes de code ... Donc, il faut avoir une plateforme qui nous permet une communication avec les encadrants en temps réel

les professeurs:

Nous avons aussi demandé l'opinion des professeurs de notre école pour étudier le besoin des deux côtés élèves et professeurs.

- Mr. BELLABDAOUI Adil:

nous avons réalisé un interview avec Mr BELLABDAOUI pour discuter les points principaux de notre projet et savoir son avis et enrichir nos idées généralement il nous a conseillé de ne pas commencer le travail à zéro et se baser sur une application existante et ajouter des fonctionnalités. Quant au

service d'endorsement c'est préférable qu'il soit limité aux professeurs pour avoir plus de confiance en cette plateforme.

- Mme BERRADA Bouchra:

Nous avons aussi réalisé un interview avec Mme BERRADA qui a aimé l'idée de notre projet et nous a donné plusieurs conseils, et a insisté sur l'importance d'avoir un service de chat et d'endorsement mais qui n'est pas limité aux professeurs mais aussi aux étudiants.

- Mr. HAMLAOUI Mahmoud

Pour notre troisième interview c'était avec Mr HAMLAOUI qui a aussi aimé notre idée et nous a proposé d'avoir une certaine distinction entre les différentes notifications et les classer selon l'ordre de l'importance

4-3 Conclusion

En analysant ces statistiques on constate donc que la plupart des étudiants utilisent facebook pour des raisons d'études alors que cette approche en réalité a une conséquence grave c'est la perte du temps car facebook ne permet pas seulement aux étudiants d'étudier mais aussi de se distraire d'où vient l'utilité de notre application qui se focalise seulement sur tout ce qui a rapport avec les études. En outre pour le cas de notre école la plupart des élèves et ainsi que les profs préfèrent avoir une plateforme unifiée pour l'échange de document et pour la communication

Ainsi on définit les besoins comme suit :

- L'application doit permettre aux étudiants de rester au courant des dernières actualités concernant leurs études (emplois du temps , plannings des contrôles...) donc on doit disposer d'un service de notification
- Elle doit aussi permettre aux étudiants de créer des groupes pour partager des documents en commun
- En outre la scolarité doit avoir la possibilité de passer les informations aux étudiants
- Elle doit permettre aux étudiants de discuter en groupes
- l'application doit être real-time app

5- Référentiel des exigences

L'approche commune à tous les domaines d'ingénierie est de définir les besoins, d'envisager des solutions, et de livrer la solution la plus appropriée. Plus précisément, les activités consistent à définir le problème, collecter les informations pertinentes, générer des idées de solutions possibles, les analyser et choisir la plus appropriée, et mettre la solution en œuvre. Les exigences forment la clé de toute de ces activités. En effet, le besoin ou le problème nécessite d'être traduit en exigences pour la solution.

5-1 Exigences fonctionnelles :

- Création des comptes pour les étudiants
- Création des comptes pour les professeurs
- Création des comptes pour les membres de la scolarité
- Création des profils pour tous les utilisateurs
- Personnalisation des profils
- Connexion
- Partage des documents avec les étudiants
- Créations des groupes selon l'année scolaire et la filière
- Créations des groupes à la demande
- Service de chat
- Service d'endorsement

- Système de recommandation
- Visualisation des profils de la part des entreprises
- Suivi des projets académique

5-2 Exigences non fonctionnelles

- Application des principes du UI/UX
- Sécurité de l'application

6- Vision stratégique :

Une vision stratégique partagée est une représentation du futur souhaitée par l'ensemble de l'entreprise. Elle implique les ressources de toute l'organisation et propose un cadre d'intervention cohérent dans la mise en œuvre des actions et d'une ambition et motivation commune. C'est comme si chaque individu dans l'entreprise prenait conscience du rôle qu'il a à tenir pour aller dans une direction connue et atteindre une destination envisagée.

La vision stratégique de notre entreprise :

- Tout d'abord on va estimer les frais du projet
- Estimer les revenus

6-1 Frais :

- Location du serveur :

Après une recherche on constate que le prix de location de serveur va nous coûter envers 30000dh la première année :

mHG-2019

Processeur : Intel Xeon Silver 4110 - 8 c / 16 t - 2.1 GHz / 3 GHz
Mémoire : 96Go
Disque : 2x1To SSD NVMe

Réseau public : À partir de 1 Gbit/s
Réseau privé : 3 Gbit/s

À partir de
2.405,99 Dhs
HT / mois

[Configurer](#)

- Test de montée en charge

Le test de montée en charge est indispensable pour nous permettre d'anticiper nos périodes d'activités stratégiques. Alors que l'utilisation augmente, nous ne pouvons pas nous permettre d'être confronté à une plateforme défaillante !

Lorsqu'il s'agit de ralentir le chargement de sites Web et d'applications, les clients s'impatientent et trouvent un remplaçant s'il ne répond pas à leurs besoins. LoadView a été créé en pensant aux performances et à l'efficacité.

Prix du LoadView par mois : 1900dh

Nous estimons que la montée en charge ne sera un problème qu'après la première année du lancement du projet et vas couter environ 23000 dh par an.

- Prix du publicité :

Nous allons choisir la publicité à travers facebook en utilisant l'approche du cout par clic qui vas nous couter environ 2dh/par click donc une estimation de 20 000 dh par an. [3]

- Estimation des charges du projet pour les premiers trois ans :

année	Montant en DHs
1 ^{ère} année	50 000
2 ^{ème} année	73 000
3 ^{ème} année	73 000

6-2 Revenus :

Nous avons choisi que l'application soit payante :

Pour que l'université crée un compte d'université et par conséquent générer des comptes pour ses étudiants elle doit payer pour chaque élève 100 dh par an

En comparaison avec le prix des réseaux sociaux professionnels comme linkedin qui coute au minimum 250dh/mois donc 2500dh /an notre application est beaucoup moins chère.

Nous estimons avoir 2000 étudiants inscrits dans notre plateforme en 1^{ère} année et 5000 en 2^{ème} année et puis 8000 en 3^{ème} année

Donc le revenu pour les trois premières années est :

année	Montant en DHs
1 ^{ère} année	200 000
2 ^{ème} année	500 000
3 ^{ème} année	800 000

6-3 Vision stratégique pour 5 ans :

Nous estimons avoir une stabilité dans les dépenses et revenus dans la 4^{ème} et 5^{ème} année parce qu'au début une application nécessite un peu de temps pour qu'elle soit utilisée par plus de gens pendant ces années on doit fasciner nos utilisateurs et gagner leur confiance afin que l'application soit plus diffusée.

Analyse

Dans cette partie nous allons établir une étude psychologique des réseaux sociaux et citer les principes du UX Design qu'on doit respecter et définir l'architecture du projet pour le moment et pour les années qui suivent

1-étude psychologique:

1-1 Comment fonctionne la psychologie des réseaux sociaux

- 1-1-1 la notion de récompense

L'utilisation des réseaux sociaux active le système hédonique du cerveau, aussi appelé système de récompense. Lors de l'utilisation de réseaux sociaux, de la dopamine est libérée. Ce neurotransmetteur est aussi connu sous le nom d'hormone du plaisir

- 1-1-2 le rôle du système de partage

La majorité des personnes interrogées déclarent partager et aimer du contenu sur les réseaux sociaux afin d'entretenir leur réseau personnel ou professionnel. Le partage est aussi un moyen d'affirmer son identité, de se mettre en valeur et de garder une image positive aux yeux de la société. Le partage a aussi lieu dans le but de transmettre des émotions. 25% des partages ont lieu en raison de l'étonnement de l'internaute face à un contenu. Les contenus humoristiques ou amusants sont aussi très populaires. La

réciprocité rentre également en compte : un partage peut être une forme de remerciement.

1-2 Discourse analysis

En se basant sur l'article de recherche :The virtual geographies of social networks: a comparative analysis of Facebook, LinkedIn and ASmallWorld publié par ZIZI PAPACHARISSI (qui est un spécialiste de la communication dont le travail a contribué à définir le domaine de la communication politique à l'ère des médias numériques contemporains. Elle est actuellement professeur et chef du département de communication à l'Université de l'Illinois à Chicago et rédactrice en chef des revues Journal of Broadcasting & Electronic Media et Social Media and Society.) à University of Illinois-Chicago, USA . On peut faire une étude comparative entre les différents réseaux sociaux en se basant sur le Discourse analysis:

Cette analyse comparative du discours porte principalement sur la manière dont les différences dans les sites de réseaux social sont communiquées à travers la conception des espaces socioculturels par le biais de la conception de l'espace en ligne. Ce faisant, il examine comment les individus adaptent ces espaces en ligne pour personnaliser et personnaliser la communication en fonction de leurs propres routines, et dans quelle mesure l'architecture en ligne leur permet de le faire. se connecter sur différents sites de réseaux sociaux et comment ce langage est-il influencé, le cas échéant, par l'architecture? Discourse analysis est utilisé pour examiner et comparer les structures de deux sites de réseautage social, chacun représentant une niche et une approche différentes du réseautage social. Mark Zuckerberg, fondateur et PDG de

Facebook, a fait remarquer qu'en concevant Facebook, il ne voulait pas que le site ait l'air `` voyant ou cool " " mais plutôt `` travail correctement" (Cassidy, 2006), produisant un `` utilitaire social qui vous connecte avec les gens qui vous entourent " (www.facebook.com, consulté le 15 août 2008) .Certaines des fonctionnalités les plus populaires de Facebook incluent le partage de photos et la possibilité pour les utilisateurs de contribuer à des applications qui fonctionnent avec la fondation open-source du site Web, rafraîchissant et rajeunissant constamment le contenu, ce qui présente un attrait pour plusieurs utilisateurs, le rendant populaire et, pour certains, addictif (Cassidy, 2006). Même si Facebook est accessible à tous, certains affirment qu'il contient un biais démographique intégré, car non seulement l'accès à Internet et l'alphabétisation sont nécessaires pour bénéficier de ses privilèges (Hargittai, 2007), mais sa base d'utilisateurs était structurée autour d'institutions éducatives privilégiées et porte donc un élément bourgeois américain (Boyd, 2007, dans Johnson, 2007). "(LinkedIn, 2007). Les utilisateurs de LinkedIn sont généralement affiliés à leur réseau de travail et utilisent le site pour maintenir une liste de coordonnées des personnes qu'ils connaissent et en qui ils ont confiance dans leur domaine d'activité, appelés" connexions ". Ce réseau de contacts est utilisé pour maintenir la communication, échanger des informations et se référer les uns aux autres. Le site utilise une `` approche d'accès sécurisé ", ce qui signifie que la connexion avec les autres nécessite soit une relation préexistante, soit l'intervention d'un contact mutuel, qui est un mécanisme conçu pour faciliter la confiance entre les membres.

1-3 conclusion:

On se basant sur le Discourse analysis et l'étude comparative on constate que la fonctionnalité d'avoir un profil personnalisé est très importante car elle permet aux utilisateurs de se distinguer des autres et par conséquent mettre en valeur leur profil. En outre la fonctionnalité d'établir des relations avec les gens est aussi importante pour cela dans notre application chaque utilisateur appartient à un ou plusieurs groupes. En outre on doit mettre en valeur la nécessité d'un système de partage et de récompense (likes)

2- UX Design

L'UX Design représente l'ensemble des émotions que peut ressentir un utilisateur par rapport à une interface donnée, que ce soit un site internet, une application mobile ou tout autre objet du quotidien.

Sur internet, les utilisateurs et leurs « expériences » ont une importance prédominante. Plus de 7 internautes sur 10 ne reviendront pas sur un site internet si leur première expérience sur celui-ci a été mauvaise. Pour exister, il

est donc nécessaire d'offrir une « expérience » plus qu'un simple produit/service.

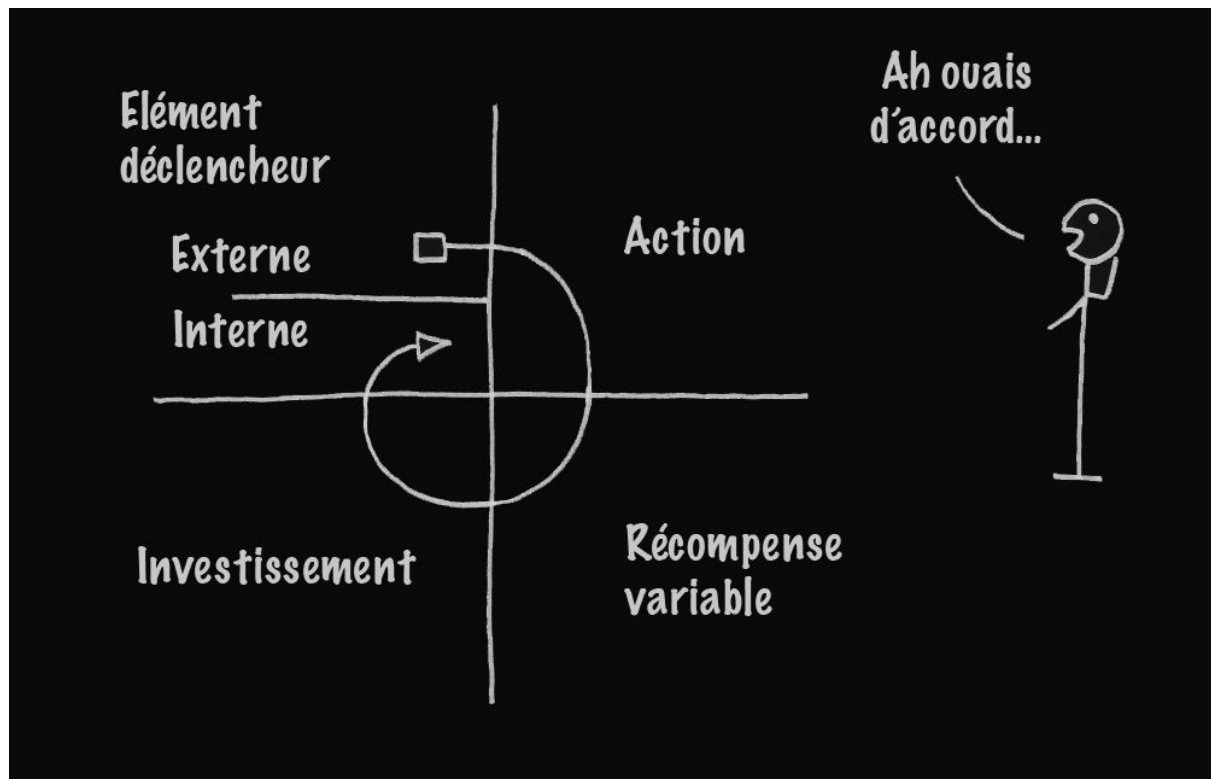
en se basant sur les principes du UX Design on doit respecter les règles suivantes dans notre projet:

- L'utilisabilité :

il désigne la capacité d'un utilisateur à utiliser un produit, à comprendre l'usage d'un produit : à quoi il sert, et comment on s'en sert. les utilisateurs de notre application sont généralement des professeurs et des étudiants des différentes universités donc ils ne sont pas forcément des informaticiens par conséquent l'application doit être facile à utiliser et intuitive

- Le hook canevas:

Développé par Nir Eyal, professeur à l'école de commerce Stanford, le modèle du crochet (Hook Canvas en anglais) suggère que pour “accrocher ses utilisateurs”, le flow de votre application doit s'articuler en 4 phases qui se complètent et se répètent formant ainsi une boucle.



dans notre cas :

- -l'élément déclencheur : différentes notifications
La peur de manquer quelque chose
- -Action: éliminer au maximum la friction
réduire la charge cognitive
- Récompense variable: likes
- Investissement : partage des documents
système d'endorsement

- adapter le design thinking:

Le Design Thinking est une méthodologie par étapes qui permet de gérer le développement d'un projet de façon innovante :

1. Empathie = se mettre à la place des autres pour comprendre un problème.
2. Définition = définir le problème à résoudre.
3. Idéation ou Brainstorming = trouver des solutions créatives.
4. Prototype = choisir la solution adaptée et concevoir une première version.
5. Test = tester le concept puis l'améliorer.

3- Architecture

3-1 architecture monolithique

Une architecture monolithique représente le modèle traditionnel unifié de conception d'un programme informatique.

Dans ce contexte, « monolithique » signifie formé d'un seul bloc. Un logiciel monolithique est conçu pour être autonome ; ses composants sont interconnectés et interdépendants plutôt qu'associés de manière flexible comme dans le cas des programmes modulaires. Dans ce type d'architecture étroitement intégrée, chaque composant et ceux qui lui sont associés doivent être présents pour permettre l'exécution ou la compilation du code.

- avantage de l'architecture monolithique

Les systèmes monolithiques offrent plusieurs avantages, en particulier en termes de gestion des frais opérationnels:

-Simplicité : Les architectures monolithiques sont simples à construire, à tester et à déployer. Les applications qui en dépendent peuvent être mises à l'échelle horizontalement, en exécutant plusieurs copies d'une application derrière un répartiteur de charge. plus précisément la mises à l'échelle horizontalement consiste à lancer l'application sur plusieurs serveurs distincts possédant

éventuellement les mêmes capacités matérielles de façon à répartir la charge de travail entre ces serveurs

- Problèmes transversaux : Avec une base de code unique, les applications monolithiques peuvent facilement gérer les problèmes transversaux, tels que l'enregistrement des logs, la configuration et le contrôle des performances

-Performances : Les composants d'une architecture unifiée partagent la mémoire vive, ce qui est plus rapide que la communication de service à service

-C'est plus efficace de commencer un projet avec une architecture monolithique car c'est moins compliqué en comparaison avec une architecture en microservices.

- les inconvénients de l'architecture monolithique

-Il dépend généralement d'une base de données unique donc on ne peut pas profiter des points forts de chaque base de données et utiliser plusieurs en même temps

- le couplage étroit: Au fil du temps, les composants deviennent étroitement liés et enchevêtrés. Ce phénomène affecte la gestion, l'évolutivité et le déploiement continu

Fiabilité : Une erreur de programmation dans l'un des modules de l'application peut la faire tomber entièrement.

Mises à jour : Parce qu'il y a une seule grosse base de code et une dépendance forte entre les composants, toute l'application doit être redéployée à chaque mise à jour.

3-2 microservices architecture

Dans une architecture à microservices, chaque composant d'application s'exécute comme son propre processus et communique avec les autres services via une API bien définie. Les microservices sont développés autour des capacités métier, et chaque service exécute une seule fonction. Les microservices peuvent être écrits à l'aide d'infrastructures et de langages de programmation différents, et vous pouvez les déployer indépendamment, comme un seul service ou comme un groupe de services.

avantages:

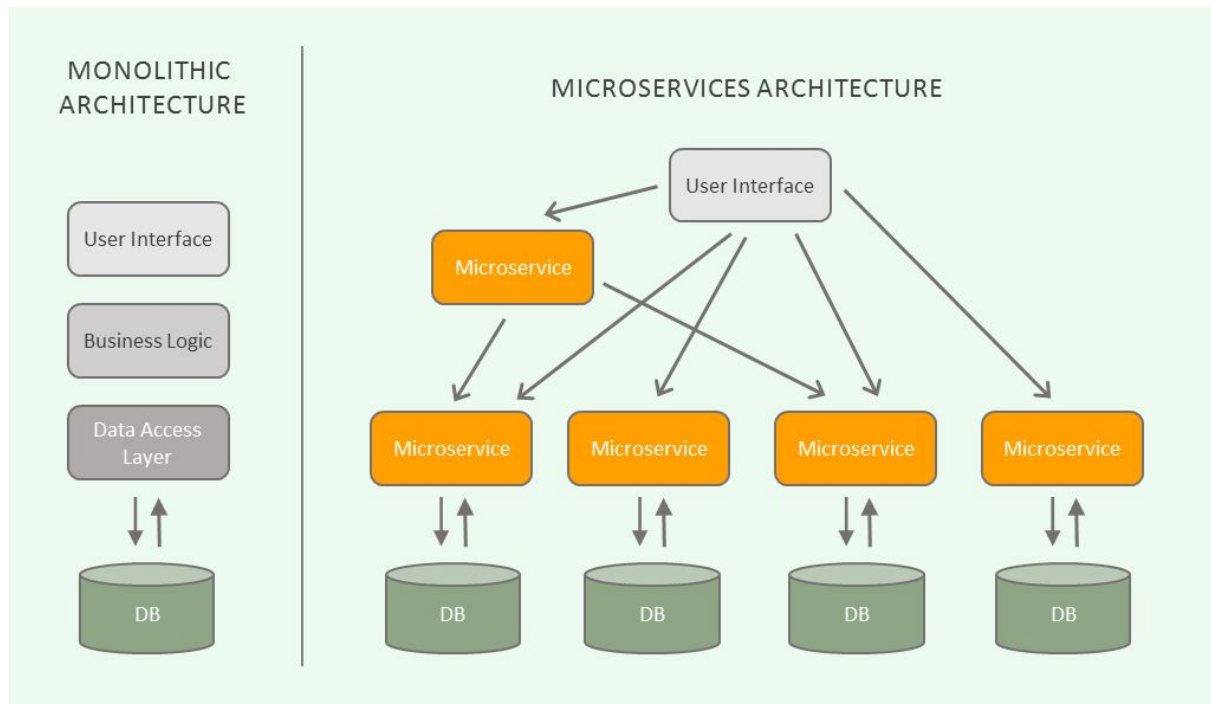
- L'architecture Microservice offre aux développeurs la liberté de développer et de déployer de manière indépendante
- Un microservice peut être développé par une équipe assez petite
- Le code pour différents services peut être écrit dans différentes langues, donc profiter des points forts de chaque langage selon le service correspondant
- Facile à comprendre et à modifier pour les développeurs, peut donc aider un nouveau membre de l'équipe à devenir productif rapidement
- Lorsqu'une modification est requise dans une certaine partie de l'application, seul le service connexe peut être modifié et redéployé: il n'est pas nécessaire de modifier et de redéployer l'application entière

- Meilleure isolation des pannes: si un microservice échoue, l'autre continuera à fonctionner
- Facile à dimensionner et à intégrer à des services tiers
- Pas d'engagement à long terme envers la pile de technologie

inconvénients:

- L'augmentation du nombre de services peut entraîner des barrières d'information
- L'architecture apporte une complexité supplémentaire car les développeurs doivent atténuer la tolérance aux pannes, et traiter différents formats de messages ainsi que l'équilibrage de charge
- Étant un système distribué, cela peut entraîner des doubles emplois
- Lorsque le nombre de services augmente, l'intégration et la gestion de produits entiers peuvent devenir compliquées
- En plus de plusieurs complexités de l'architecture monolithique, les développeurs doivent faire face à la complexité supplémentaire d'un système distribué
- Les développeurs doivent mettre des efforts supplémentaires dans la mise en œuvre du mécanisme de communication entre les services
- La gestion des cas d'utilisation qui couvrent plus d'un service sans utiliser de transactions distribuées n'est pas seulement difficile, mais nécessite également une communication et une coopération entre différentes équipes
- L'architecture entraîne généralement une consommation de mémoire accrue

3-3 monolithique vs microservices



3-4 Décision:

après avoir étudié les avantages et les inconvénients de chaque type d'architecture on choisit pour le moment une architecture monolithique car:

-notre équipe est trop petite et nous n'avons pas les moyens pour recruter plusieurs développeurs et les payer..

-c'est plus cher et coûteux au niveau de déploiement

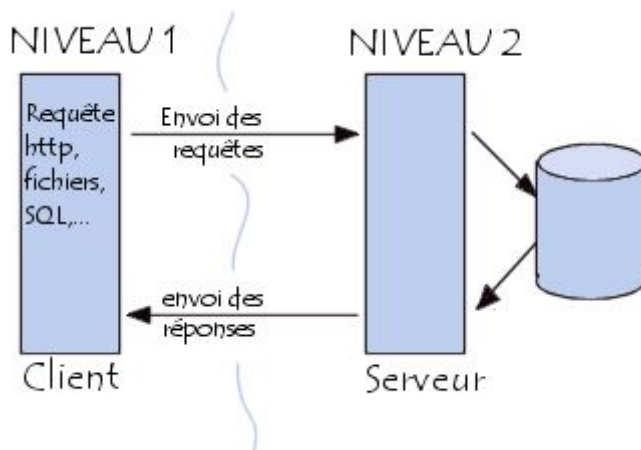
-nous n'avons pas un grand nombre d'utilisateurs pour le moment nous avons conçu cette application pour l'ensias mais prochainement nous voudrions bien la généraliser.

-c'est plus difficile de faire une bonne conception de l'application toute entière en microservices au début un essaie avec une architecture monolithique au début sera plus efficace

4- architecture monolithique de l'application:

4-1 Architecture 2 tiers

Nous avons choisi d'utiliser une architecture à deux niveaux (aussi appelée *architecture 2-tier*, *tier* signifiant *rangée* en anglais) qui caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.



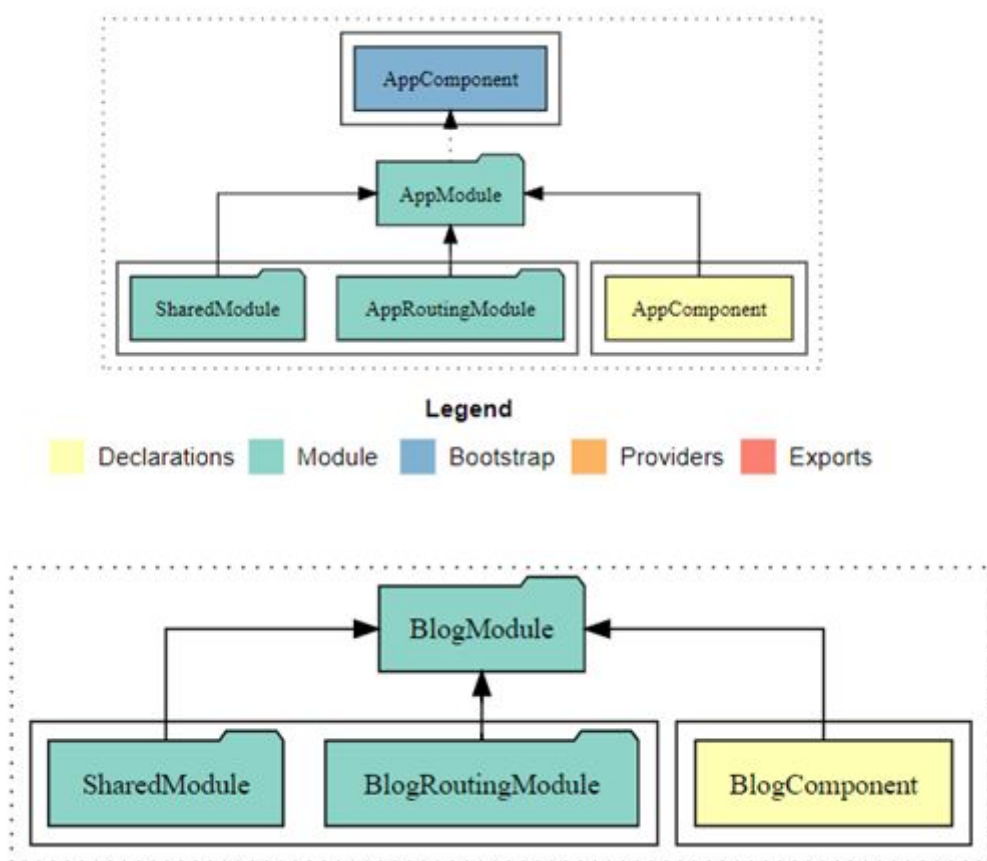
4-2 Architecture modulaire

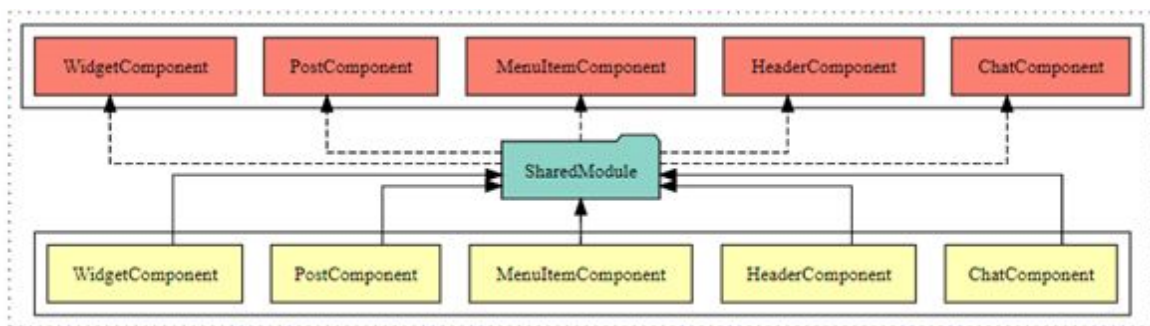
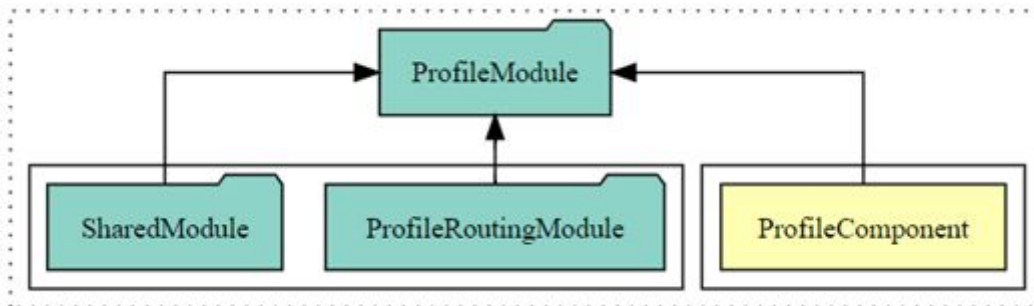
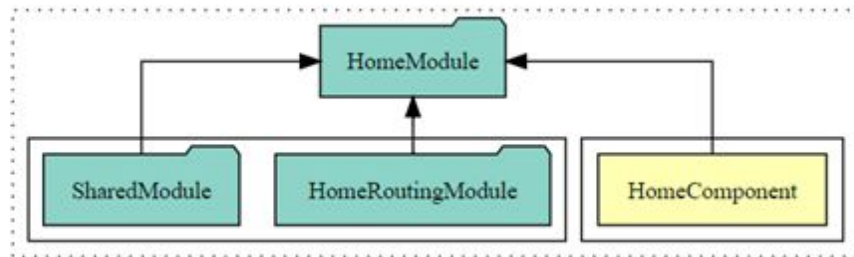
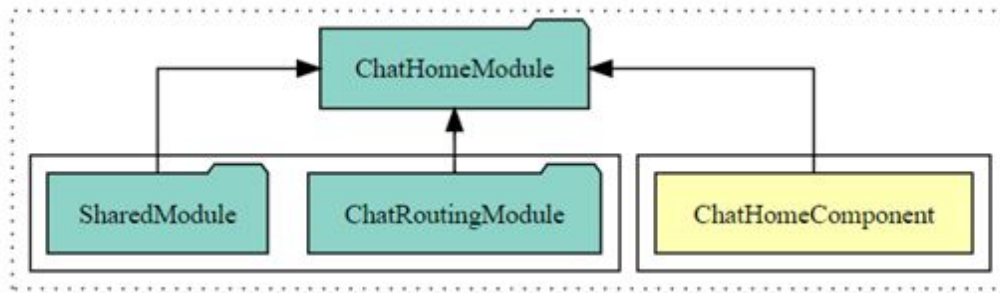
La conception modulaire ou «modularité dans la conception» est une approche de conception qui subdivise un système en parties plus petites appelées modules ou patins qui peuvent être créés indépendamment et ensuite utilisés dans différents systèmes. Un système modulaire se caractérise par un partitionnement fonctionnel en modules discrets évolutifs et réutilisables, une utilisation rigoureuse d'interfaces modulaires bien définies et l'utilisation de normes industrielles pour les interfaces.

Les avantages:

- flexibilité dans la conception: compose des modules totalement indépendants et réutilisables.
- réduction des coûts: les modules ne sont pas chargés au moment de l'exécution mais jusqu'à ce qu'ils soient injectés.

Les modules de notre projet:





5- architecture en microservices prochainement:

5-1 Exemple des entreprises utilisant architecture Microservices:

- Netflix
- Amazon
- Uber
- Ebay

Netflix a une architecture répandue qui a évolué du monolithique au SOA. Il reçoit plus d'un milliard d'appels chaque jour, à partir de plus de 800 types d'appareils différents, vers son API de streaming vidéo. Chaque appel API invite ensuite environ cinq appels supplémentaires au service backend.

Amazon a également migré vers des microservices. Ils reçoivent d'innombrables appels d'une variété d'applications, y compris des applications qui gèrent l'API de service Web ainsi que le site Web lui-même, ce qui aurait été tout simplement impossible à gérer pour leur ancienne architecture à deux niveaux.

Le site d'enchères eBay est un autre exemple qui a connu la même transition. Leur application principale comprend plusieurs applications autonomes, chacune exécutant la logique métier pour différents domaines fonctionnels.

5-2 les patterns de la décomposition

Un autre défi consiste à décider comment partitionner le système en microservices. C'est vraiment un art, mais il existe un certain nombre de stratégies qui peuvent aider:

- Décomposer par capacité métier et définir les services correspondant aux capacités métier.
- Décomposer par sous-domaine de conception piloté par domaine.
- Décomposer par verbe ou cas d'utilisation et définir les services qui sont responsables d'actions particulières. par exemple. un service d'expédition qui est responsable de l'expédition des commandes complètes.
- Décomposer par par noms ou ressources en définissant un service qui est responsable de toutes les opérations sur les entités / ressources d'un type donné. par exemple. un service de compte qui est responsable de la gestion des comptes d'utilisateurs.

5-3 les microservices et l'apprentissage automatique

À mesure que l'apprentissage automatique entre en production, il devient de plus en plus clair qu'une architecture de microservices peut convenir à ce type d'application. Il y a deux raisons principales à cela:

Après la formation de vos modèles, l'inférence est généralement sans état, car aucune donnée ni aucun état ne doivent être conservés, les services indépendants fonctionnent

L'apprentissage automatique est un processus à forte intensité de calcul qui nécessite souvent du matériel spécialisé (comme les GPU), et vous ne voulez pas que cela fasse partie intégrante des exigences de votre serveur

5-4 microservice ecosystem



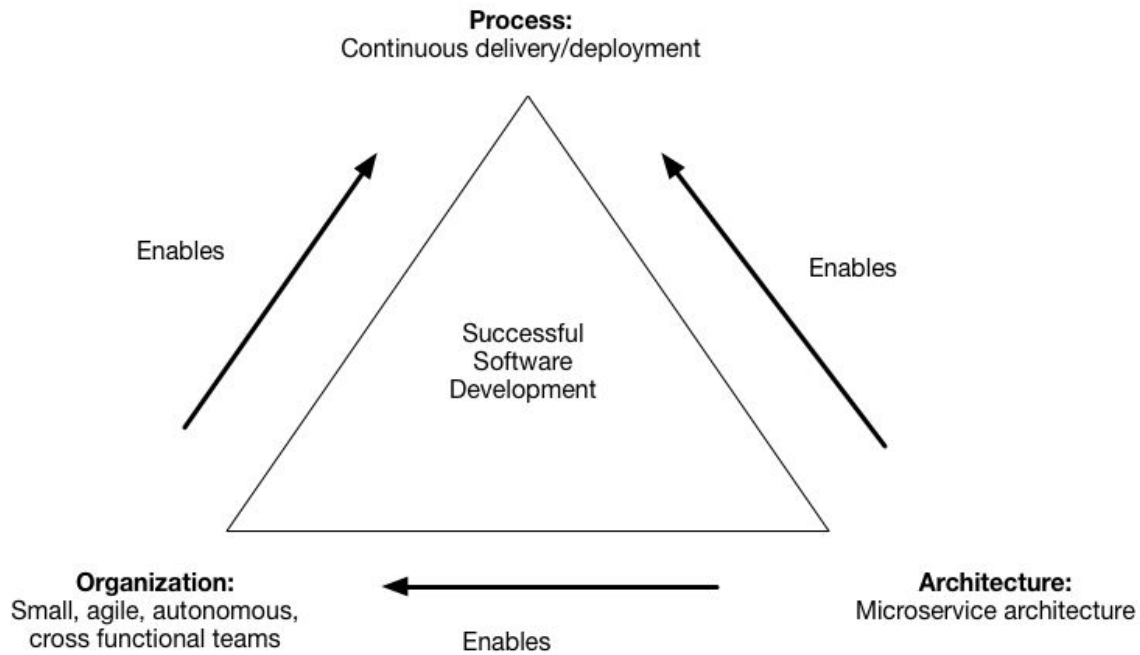
L'écosystème de microservices est une plate-forme de services qui encapsule chacun une capacité commerciale. Une capacité commerciale représente ce qu'une entreprise fait dans un domaine particulier pour remplir ses objectifs et ses responsabilités. Chaque microservice expose une API que les développeurs peuvent découvrir et utiliser de manière libre-service. Les microservices ont un cycle de vie indépendant. Les développeurs peuvent créer, tester et publier chaque microservice indépendamment. L'écosystème de microservices impose une structure organisationnelle d'équipes autonomes de longue date, chacune responsable d'un ou de plusieurs services.

5-5 la démarche de la décomposition choisi pour notre application

comme indiqué précédemment il existe plusieurs méthodes de décomposition nous avons choisi la décomposition par capacité métier

Une capacité métier est un concept issu de la modélisation de l'architecture métier. Une capacité commerciale est quelque chose qu'une entreprise fait pour générer de la valeur. L'ensemble des capacités pour une entreprise donnée dépend du type d'entreprise. Par exemple, les capacités d'une compagnie d'assurance incluent généralement les ventes, le marketing, la souscription, le traitement des réclamations, la facturation, la conformité, etc. Les capacités d'une boutique en ligne incluent les ventes, le marketing, la prise de commandes, la gestion des stocks et l'expédition en utilisant quelque chose comme l'inventaire xero gestion et autres logiciels de gestion des stocks.

Le but de l'architecture de microservices est d'accélérer le développement logiciel en permettant une livraison / un déploiement continu.



étapes de la décomposition

1-définir les capacités métiers

Les capacités métiers d'une organisation capturent ce qu'est l'entreprise. Ils sont généralement stables. En revanche, la façon dont une organisation mène ses activités évolue au fil du temps, parfois de manière spectaculaire. Cela est particulièrement vrai aujourd'hui, avec l'utilisation croissante de la technologie pour automatiser de nombreux processus métier

2-définir les services

Une fois que vous avez identifié les capacités commerciales, vous définissez ensuite un service pour chaque capacité ou groupe de capacités

connexes. Pour certaines entreprises, elles constatent que l'utilisation d'un logiciel de bon de travail peut être un bon moyen de suivre les commandes des clients et des travailleurs. Un avantage clé de l'organisation des services autour de capacités, qui sont stables, est que l'architecture résultante sera relativement stable. Les composants individuels de l'architecture peuvent évoluer à mesure que les aspects de l'entreprise changent, mais l'architecture elle-même reste inchangée

3-UTILISER DES SCÉNARIOS POUR DÉTERMINER LA COLLABORATION DES SERVICES

L'architecture d'une application comprend à la fois des éléments logiciels (les services) et les relations entre eux (mécanismes de communication). Par conséquent, après avoir identifié les services, nous devons décider comment les services communiquent. Pour ce faire, nous devons considérer comment les services collaborent au cours de chaque scénario. Les opérations système définissent les scénarios et pilotent la définition de l'architecture. Étant donné qu'une opération système est une demande du monde extérieur, la première décision à prendre est de savoir quel service traite initialement la demande. Après cela, nous devons décider quels autres services sont impliqués dans le traitement de la demande et comment ils communiquent.

4-ATTRIBUTION DES OPÉRATIONS DU SYSTÈME AUX SERVICES

La première étape consiste à décider quel service est le point d'entrée initial d'une demande. De nombreuses opérations système sont parfaitement mappées à un service, mais parfois le mappage est moins évident

5-DÉTERMINER LA COLLABORATION DES SERVICES

Certaines opérations système sont entièrement gérées par un seul service. D'autres opérations système couvrent plusieurs services. Les connaissances nécessaires pour gérer l'une de ces demandes peuvent, par exemple, être dispersées autour de plusieurs services

5-6 application de la démarche : La décomposition de notre application

5-6-1 les capacités métiers:

- user info
- chat
- notification
- recommendation
- school statistiques
- doc uploading
- sharing
- endorsement
- relations management
- authentication and acces management

5-6-2 les microservices de notre application et les langages de développement et les bases de données

- **chat:**

language : Erlang est un langage de programmation, supportant plusieurs paradigmes : concurrent, temps réel, distribué. utilisé par whatsapp

base de données :Mnesia qui est est un système de gestion de base de données distribué et en temps réel, écrit dans le langage de programmation Erlang. utilisée par whatsapp

- **user info microservice/ doc uploading microservice/ sharing microservice**

language: javascript à l'aide de nodejs

base de données: mongodb qui est nosql database à base de documents

- **notification**

language: javascript en utilisant nodejs

database:Mnesia car elle est en temps-réel

- **relations management**

language: javascript en utilisant nodejs

database: postgresql

- **recommendation**

language: python

database: le type de base de données le plus utilisé est graph database par exemple Neo4j

- **statistiques :**

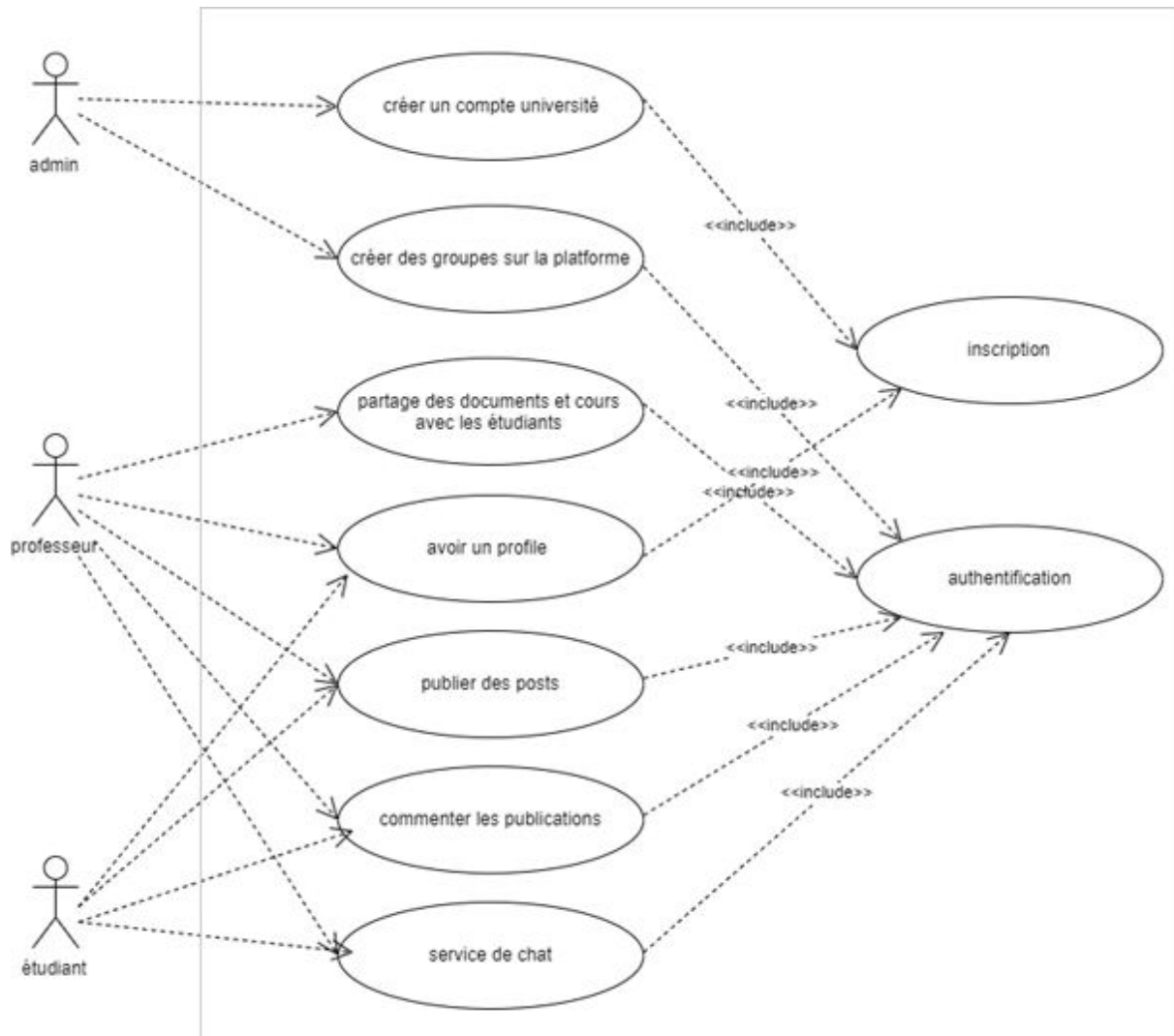
language : Il existe de nombreuses autres petites langues qui sont utilisées dans certaines parties du monde universitaire ou de l'industrie lorsque des mathématiques complexes sont nécessaires. L'un de ces langages est R, un langage de programmation largement utilisé en statistique et en analyse de données. R a quelques fonctionnalités utiles pour de tels secteurs, comme son propre format de documentation et le fait que de nombreuses fonctions standard sont écrites dans R lui-même, donc elles sont faciles à analyser pour leurs utilisateurs. Mais le meilleur avantage est la grande disponibilité des bibliothèques et de l'expertise pour les statistiques et l'analyse des données

database : postgresql

Conception

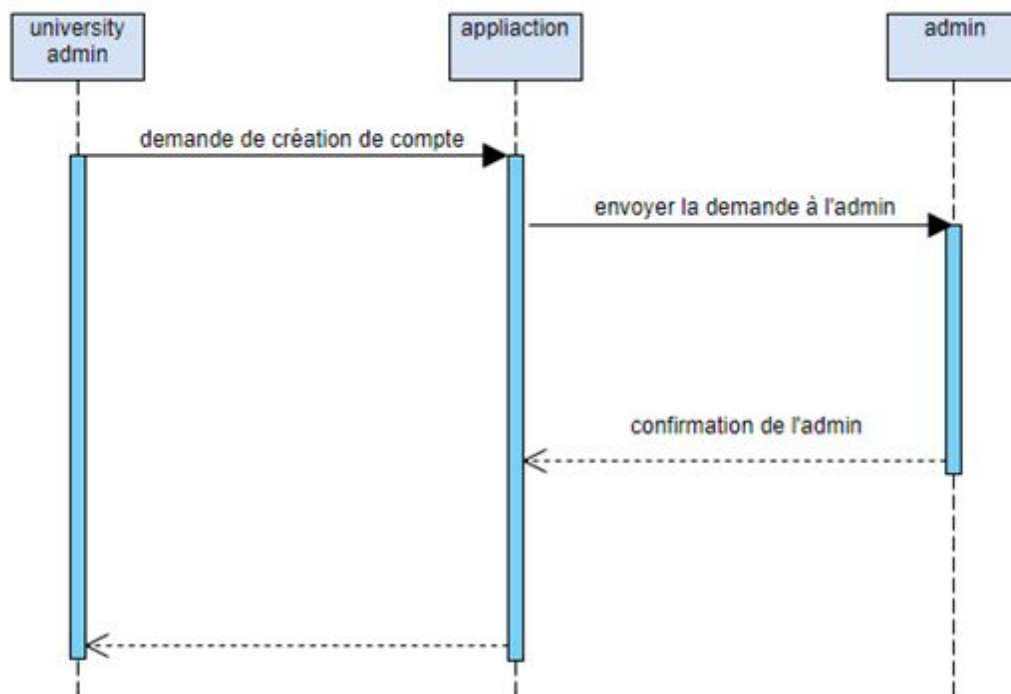
Dans cette partie nous allons présenter le diagramme use case, diagramme d'activité, et diagramme entité-relation.

1- diagramme use case

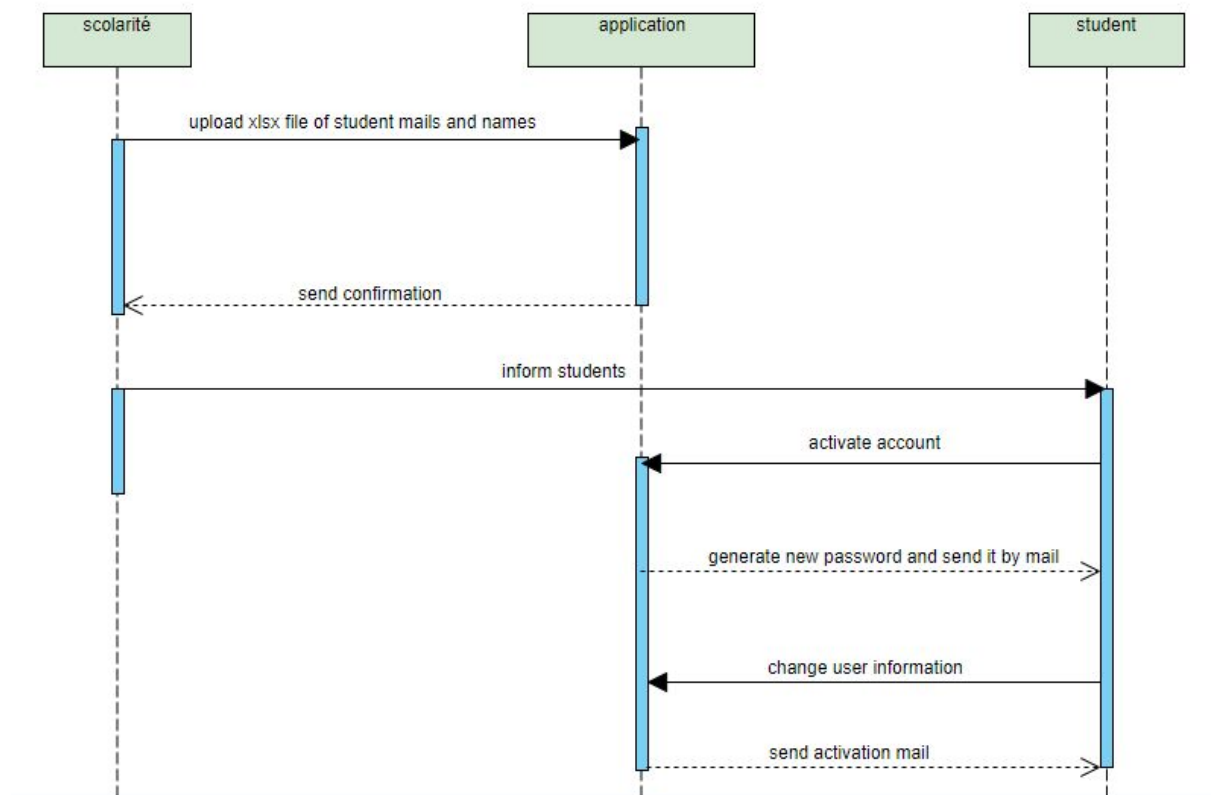


2- diagramme de séquence

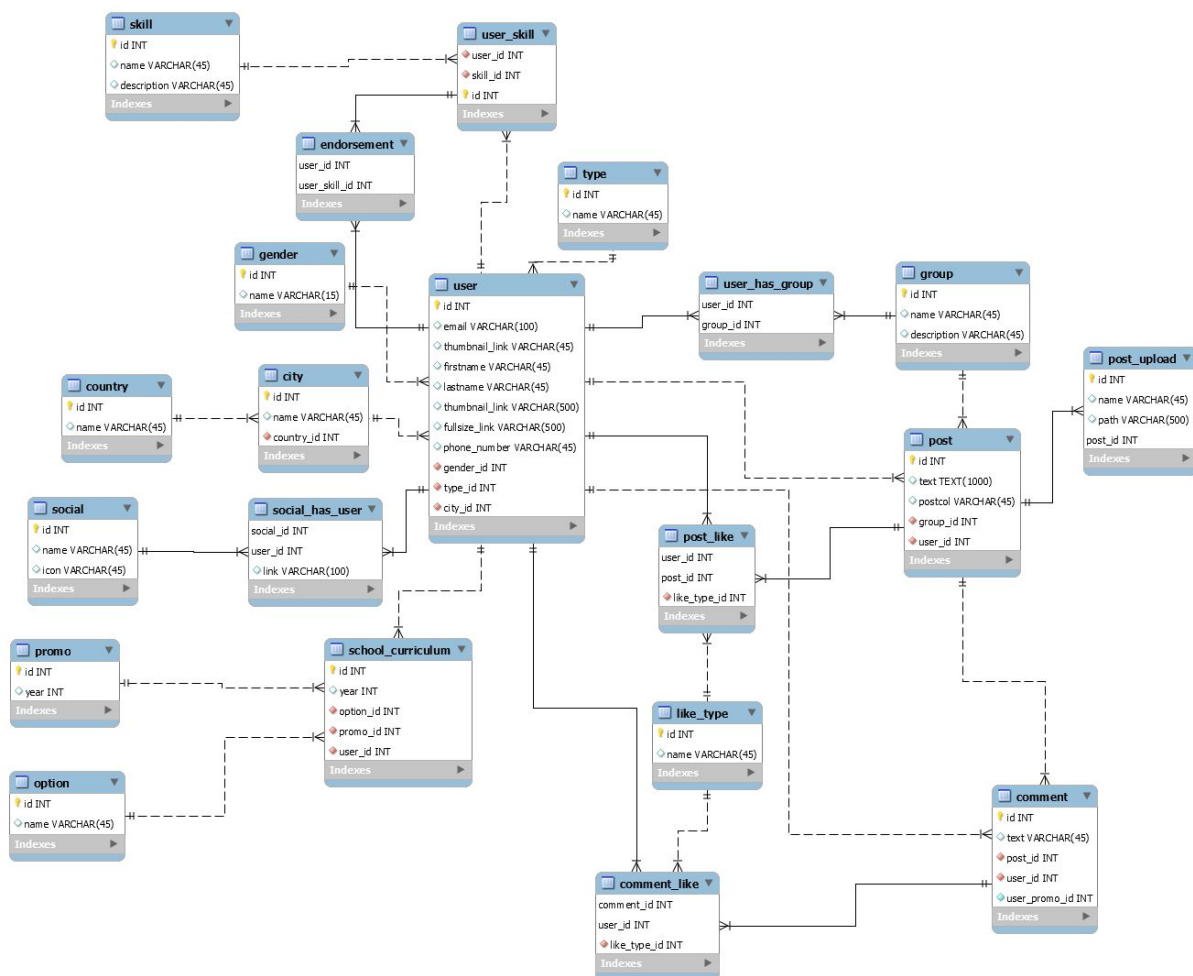
2-1 création de compte de l'université



2-2 génération des comptes des étudiants



3- diagramme entité relation



Réalisation

Dans cette partie nous allons aborder les différentes technologies utilisées ainsi que des screenshots de la réalisation

1-outils

1-1-backend

1-1 nodejs



Ce sera ici un serveur Web (bien qu'il peut être bien plus que ça). Il va s'occuper de recevoir les requêtes HTTP, de les traiter et de renvoyer un résultat. C'est donc l'équivalent d'Apache par exemple. On écrit le code en Javascript.

1-2 express



C'est un framework web pour Node.js et écrit en Node.js. C'est son framework le plus célèbre. Il vous simplifie grandement la vie à l'aide de fonctions vous évitant de nombreuses lignes répétitives de code.

1-2 frontend

Angular:



Réalisé par Google, c'est un framework front-end. Ce qui signifie qu'il est utilisé côté client, et non sur votre serveur. C'est lui qui va gérer ce qui s'affiche sur la page de l'utilisateur

1-3 base de données

1-postgresql:



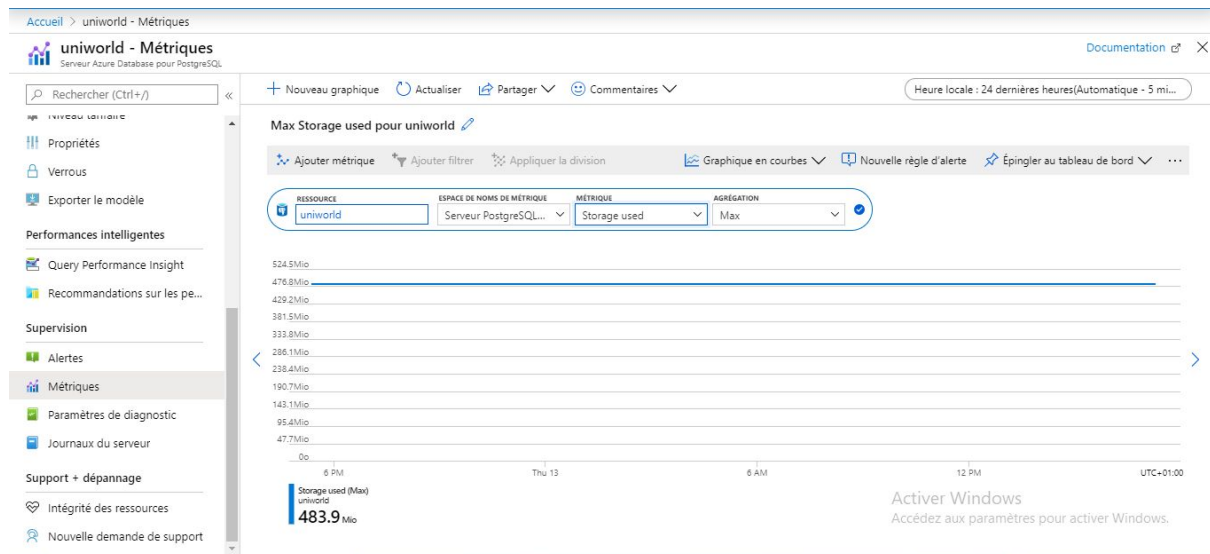
nous avons choisi une base de données relationnel car elle serait la meilleure recommandation, car il aura la fiabilité des données et les outils relationnels nécessaires pour gérer les interactions et les relations entre de nombreux utilisateurs. En outre Postgresql et parmi les meilleurs base de données relationnel car :

- elle a un excellent support des contraintes
- De nombreux outils tiers sont disponibles
- Développé activement
- elle a une communauté forte
- une bonne documentation

2-hebergement de la base de donnée

Nous avons déployé la base de données sur Azure. En effet notre mail d'université nous offre un pack de 100 euro nous avons utilisé 50 euro pour deployer la base de donnée postgresql sur Azure

3- problèmes rencontrés lors de l'hébergement



l'espace de stockage affiché dans AZURE est presque 500 Mo alors qu'en réalité l'espace de stockage ne dépasse pas 15 Mo

127.0.0.1:49772/browser/

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents public.theuser/...

Statistics	Value
Blocks read	618
Blocks hit	402328
Tuples returned	5808056
Tuples fetched	78398
Tuples inserted	3
Tuples updated	69
Tuples deleted	0
Last statistics reset	2020-02-10 21:14:29.223763+00
Tablespace conflicts	0
Lock conflicts	0
Snapshot conflicts	0
Bufferpin conflicts	0
Deadlock conflicts	0
Temporary files	0
Size of temporary files	0 bytes
Deadlocks	0
Block read time	8928.783
Block write time	0
Size	10077 kB

Activer Windows
Accédez aux paramètres pour activer Windows.

1-4 quelque libraries utilisées

1-sendgrid

encore une fois grâce au mal de l'université nous avons accès à un student pack contenant une version premium du sendgrid. SendGrid fournit un service basé sur le cloud qui aide les entreprises à envoyer leurs e-mails.

Nous avons créé un compte et généré une api Key mais malheureusement le compte c'est bloqué le lendemain

afin de résoudre le problème nous l'avons poster sur github et nous avons reçu une réponse de la part des contributeurs de la librairie indiquant que le problème est lié au support team

17:51



Re: [sendgrid/sendgrid-python]
You are not authorized to access
SendGrid, please contact
Support. (#806)



Boîte de réception



childish-sambino Avant-hier

à sendgrid/sendgrid-python, m...



This is not something that can be fixed in this library and is an issue that only support can resolve. Please submit a request for login support here:

<https://support.sendgrid.com/hc/en-us/requests/new#login-issue>

—

You are receiving this because you commented. Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).



Répondre



Répondre à tous



Transférer



finalement nous avons constaté que plusieurs utilisateurs de l'application avaient le même problème

2- nodemailer-sendgrid-transport

c'est un plugin de transport pour Nodemailer qui permet d'envoyer via l'API Web de SendGrid!

3- nodemailer

Nodemailer est un module pour les applications Node.js qui permet d'envoyer facilement des e-mails . Le projet a démarré en 2010 alors qu'il n'y avait pas d'option raisonnable pour envoyer des e-mails, c'est aujourd'hui la solution vers laquelle la plupart des utilisateurs de Node.js se tournent par défaut.

4- multer

Multer est un middleware node.js pour gérer les données en plusieurs parties / formulaires, qui est principalement utilisé pour télécharger des fichiers.

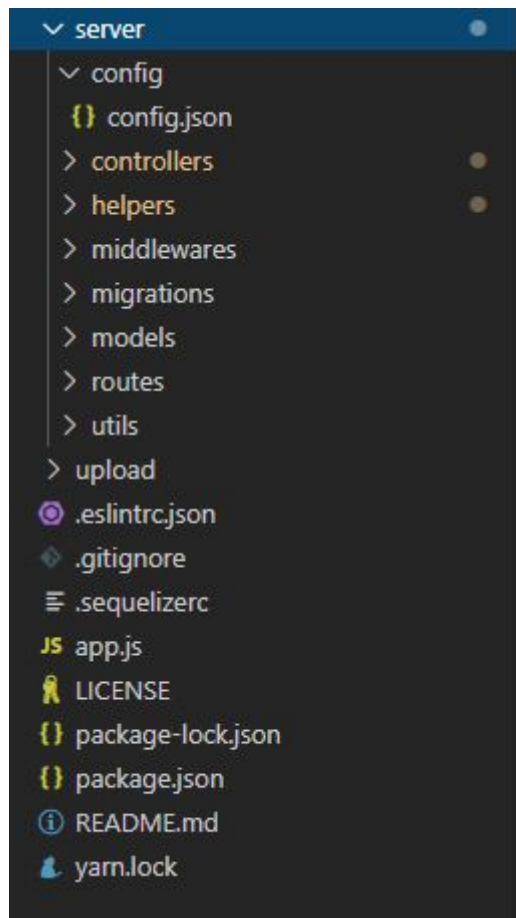
5- xlsx

parser et writer pour différents formats de feuille de calcul. Implémentation de la salle blanche Pure-JS à partir des spécifications officielles, des documents associés et des fichiers de test. Accent mis sur l'analyse et la robustesse de l'écriture, la compatibilité des fonctionnalités inter-formats avec une représentation JS unifiée et la compatibilité du navigateur ES3 / ES5 avec IE6.

6- morgan

Morgan est utilisé pour enregistrer les détails du request

2- Architecture du code



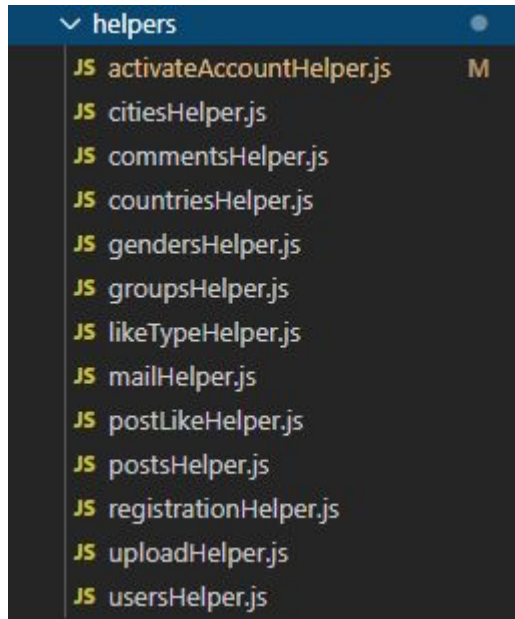
2-1 models

▼ models
JS city.js
JS comment_like.js
JS comment.js
JS country.js
JS endorsement.js
JS gender.js
JS group.js
JS index.js
JS like_type.js
JS option.js
JS post_like.js
JS post_upload.js
JS post.js
JS promo.js
JS school_curriculum.js
JS skill.js
JS social_has_user.js
JS social.js
JS type.js
JS uniworld.js
JS user_groups.js
JS user_skill.js
JS user.js

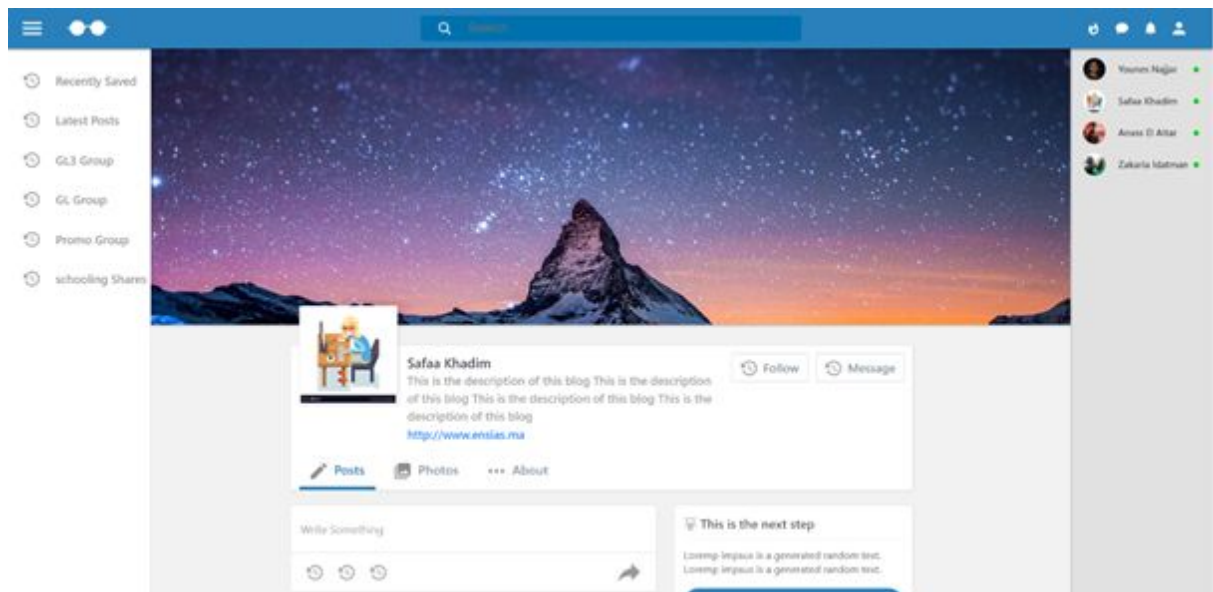
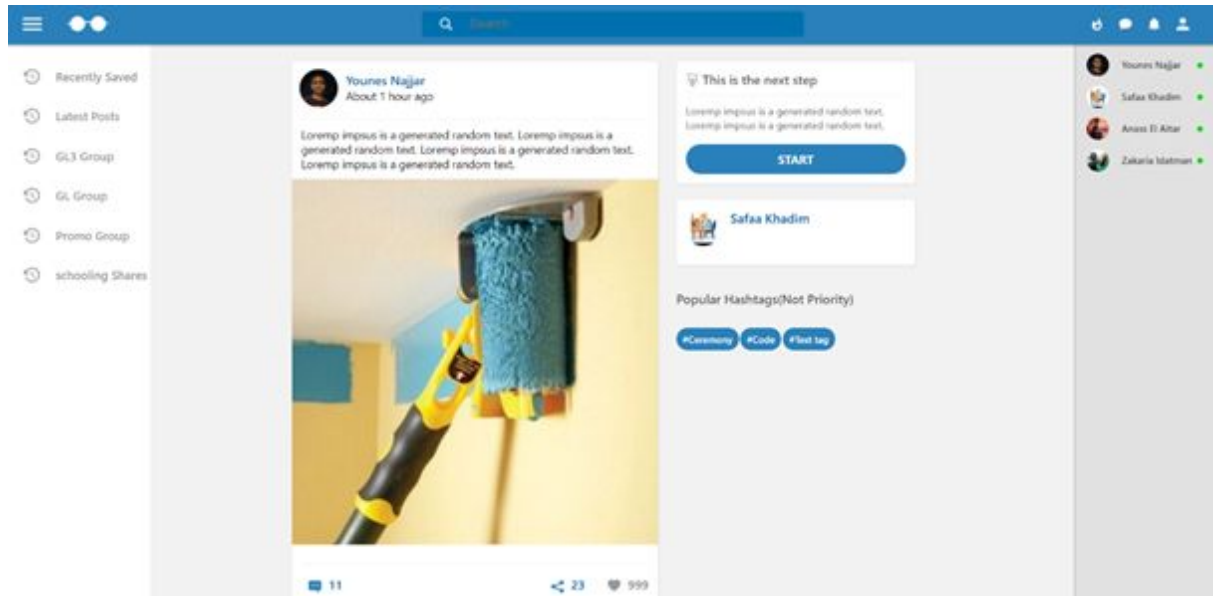
2-2 controllers

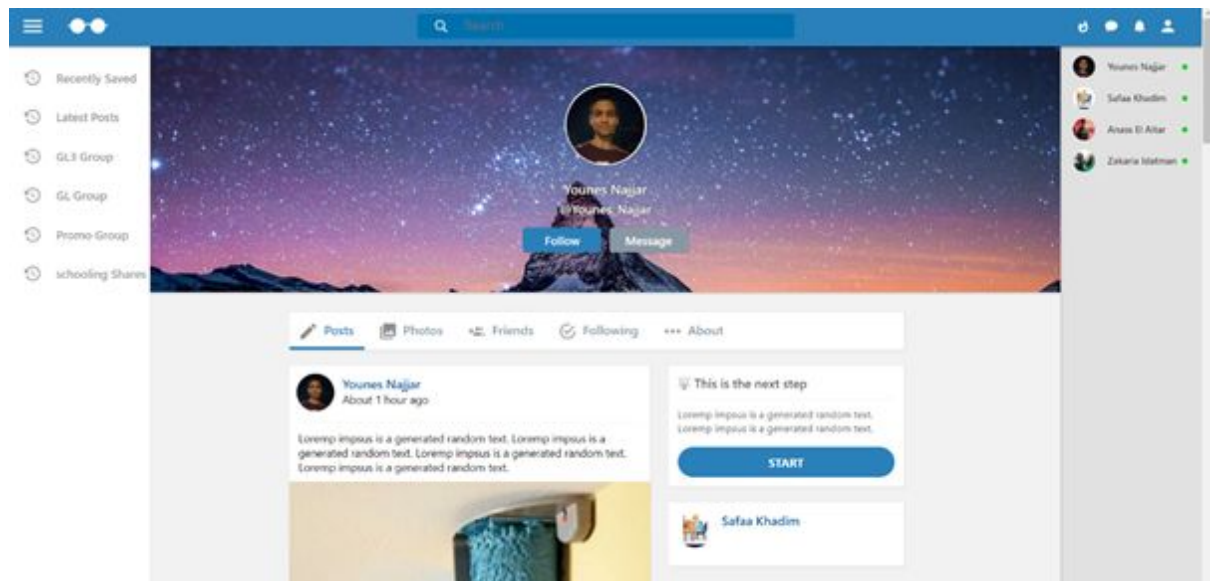
▼ controllers
JS authentication.js
JS cities.js
JS comments.js
JS countries.js
JS firstLogin.js
JS genders.js
JS groups.js
JS home.js
JS index.js
JS posts.js
JS users.js

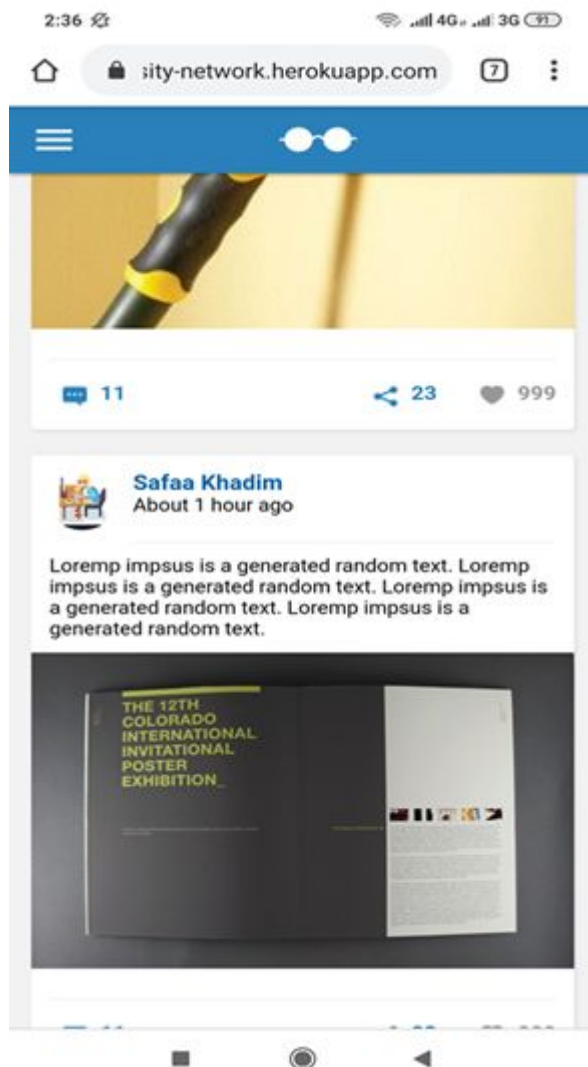
2-3 helpers

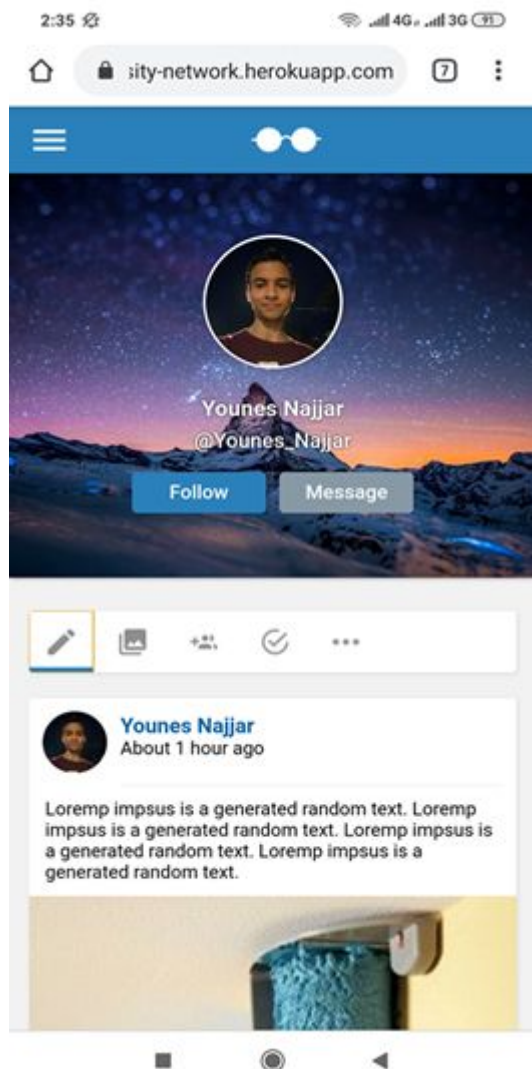


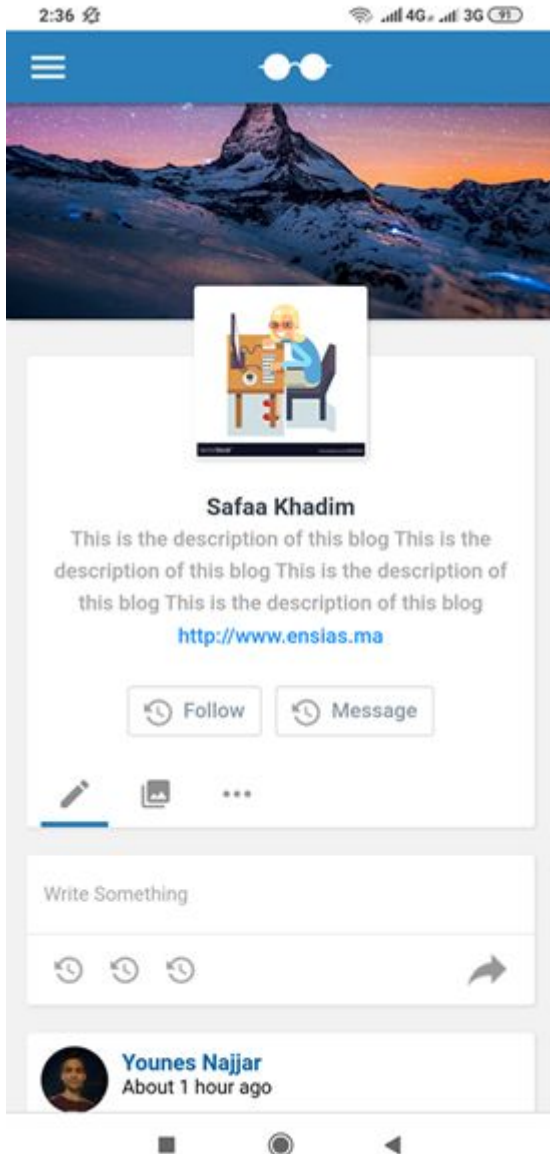
3-Screenshots

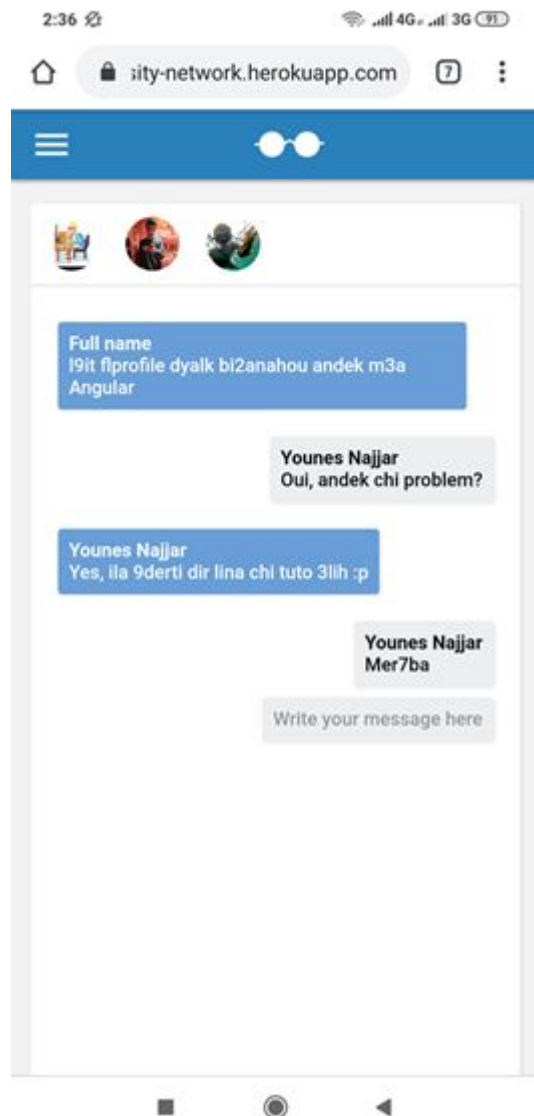












Bibliographie

<https://www.lemagit.fr/tribune/Microservices-avantages-et-inconvenients-de-la-nouvelle-structure-de-serveurs-web>

<https://www.g1site.com/psychologie-partages-reseaux-sociaux/>

<https://livebook.manning.com/book/microservices-patterns/chapter-2/8>

<https://martinfowler.com/articles/break-monolith-into-microservices.html>

<https://medium.com/code-well-live-forever/emails-with-nodemailer-sendgrid-c98cd37c8e03>

<https://laracasts.com/discuss/channels/general-discussion/has-anyone-else-had-issues-with-sendgrid-can-you-recommend-some-alternatives>

<https://medium.com/@madhukaudantha/microservice-architecture-and-design-patterns-for-microservices-e0e5013fd58a>

<https://www.slant.co/topics/6224/~databases-for-building-social-network-like-apps>

<https://www.morpheusdata.com/blog/2015-04-01-mysql-vs-mongodb-the-pros-and-cons-when-building-a-social-network>

<https://www.researchgate.net/>

<https://www.tutorialspoint.com/>

<https://www.quora.com/>

<http://gen.lib.rus.ec/>

