

## Présentation

QuoidNeuf est un système de messagerie WEB permettant aux utilisateurs de s'envoyer des messages et des photos. Une fois son compte créé, l'utilisateur peut :

- personnaliser son profil
- ajouter des contacts issus des autres abonnés
- créer des groupes d'utilisateurs
- envoyer des messages ou des photos aux contacts ou aux groupes
- Visualiser les anciens messages déjà envoyés à ce contact ou groupe

Cette application doit être conçue en mode “responsive design”, et donc utilisable sur smartphone via le navigateur de l'appareil. Afin d'avoir le choix, deux feuilles de styles seront fournies, l'une fidèle aux couleurs et logos de l'université, l'autre au choix

1. Proposer un MCD conforme aux besoins de cette application.
2. Réaliser un script SQL contenant la création des tables issues du MCD ainsi que les insertions nécessaires au scénario précédent
3. Ajouter à votre script un jeu de données permettant de tester l'application (plusieurs utilisateurs, plusieurs contacts, plusieurs groupes, plusieurs messages déjà envoyés)
4. Réaliser une ébauche des pages principales en vous appuyant sur les requêtes précédentes.
5. Ajouter les pages secondaires, CSS
6. Ajouter l'authentification, les Pools et fonctionnalités avancées.
7. Faire des tests de montée en charge du site à l'aide de scripts de génération de gros volumes de données, de tests JMETER etc ...

## Travail à faire

1. Le travail peut être réalisé en binôme, mais les responsabilités doivent être établies. A la fin, une feuille des tâches avec son responsable doit être impérativement rendue.
  2. Etablir le MCD et la base permettant de gérer cette application. C'est le coeur du problème ! Comment stocker l'information ? comment l'interroger ?
  3. Faites sobre et modeste au départ. inutile de faire joli ... l'important est déjà d'avoir une base et possibilité de déclarer des places et de rechercher des places à travers des pages austère mais fonctionnelle pour que chacun puisse travailler ! Tout le reste peut être complété par la suite ... Authentification, notation, CSS, Pool, pages Amis et Admin, SSL, etc ...
- 
1. Une documentation décrivant le projet réalisé, les choix faits et les technologies employées devra être fournie en PDF. Elle comprendra notamment 4 parties totalement indépendantes :
    - (a) Déploiement. Décrit l'installation et le déploiement de ce projet. Plus cette documentation sera courte, mieux ce sera. Notamment, le déploiement sur une machine donnée, avec un nom de contexte donné et un nom de base donné doit nécessiter le moins de travail possible.
    - (b) Synthèse
      - i. Objectif du travail
      - ii. Ce qui a été fait
      - iii. Les améliorations à apporter
      - iv. En quoi ce projet était difficile
      - v. Conclusion

- (c) La description technique du projet
  - i. Objectif techniques du travail
  - ii. Principe de réalisation
  - iii. Difficultés techniques rencontrées et solutions apportées.
  - iv. Conclusion
- (d) La documentation utilisateur (tutorial)
  - i. Objectif de cette application
  - ii. Du point de vue de l'utilisateur

## Contrôle du travail

Le projet doit tourner en salle TP et vous devez pouvoir en faire une démonstration. L'attention sera particulièrement portée sur les technologies que vous mettrez en oeuvre pour réaliser ce site, notamment :

- L'ergonomie du site,
- Mise en place d'une architecture MVC avec minimum de code dans les vues JSP
- Utilisation des EL-expressions voire de la JSTL
- Code coté client vérifiant autant que possible les informations saisies (JavaScript)
- Feuilles de styles externalisées (CSS, ...)
- Eviter l'injection SQL
- Entêtes mises en commun (fichiers includes, les logos DA2I et Lille1 doivent figurer sur toutes les pages ...)
- Système de trace d'activité du serveur (Logger, Valve...)
- Système d'authentification (Realm, ....)
- Accès à la base via un pool de connexions
- Cryptage des informations via un serveur sécurisé (SSL,...)
- Paramètres dans web.xml pour tout ce qui pourrait changer (logo, nom de la formation, base de données etc ..)
- Validation HTML des pages par Tidy ou par le W3C. Présence de l'icône de test sur chaque page.

Une fois cela réalisé, rendre sur Moodle une archive portant vos noms (ex DurandDupont.zip) et contenant la doc, un fichier README pour le déploiement, les scripts SQL nécessaires et le WAR de votre application.