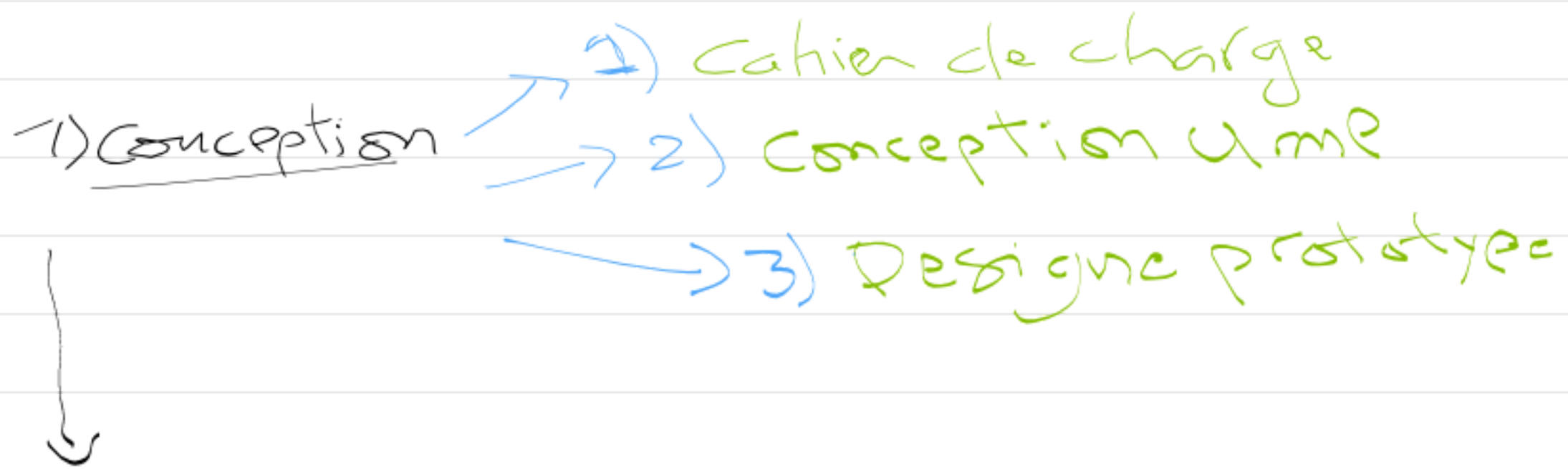
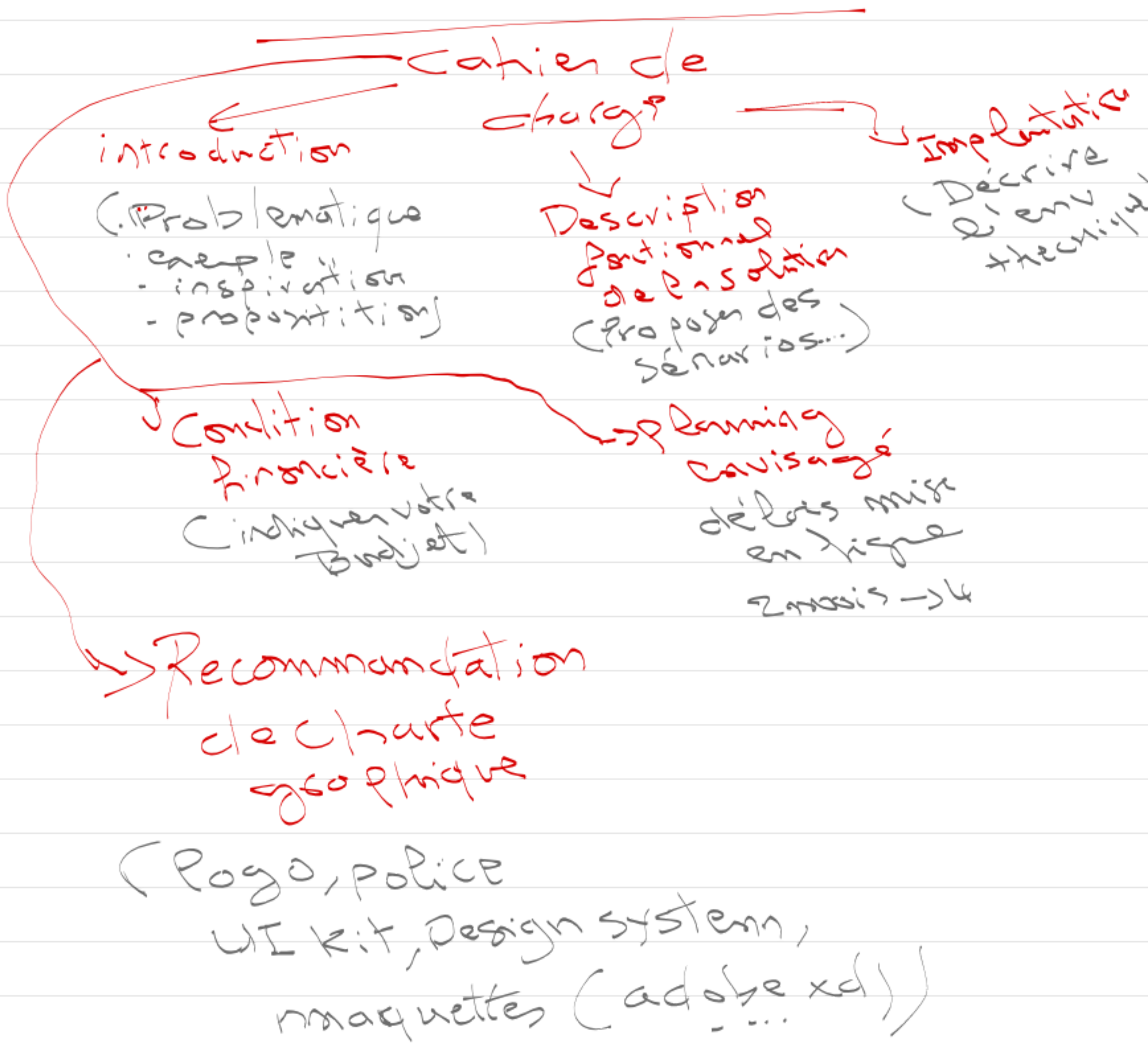


Todo List: (SPA)



2) Backend < —————> 3) Frontend



UML conception:

1) Diagramme de cas d'utilisation:

étape 1) Identifier les Utilisateurs:

Dev
Développeur

Lead
Chef

Manager

étape 2) cas d'utilisation pour chaque user

Dev → Ajouter une tâche
→ Supprimer "
→ modifier une tâche
→ Consulter list des tâches
→ Filter par titre, status les tâches

Lead → Consulter list des développeurs avec leur tâches
→ ...

Manager → crud Lead
→ Consulter l'état
→ crud projet d'avancement des projets

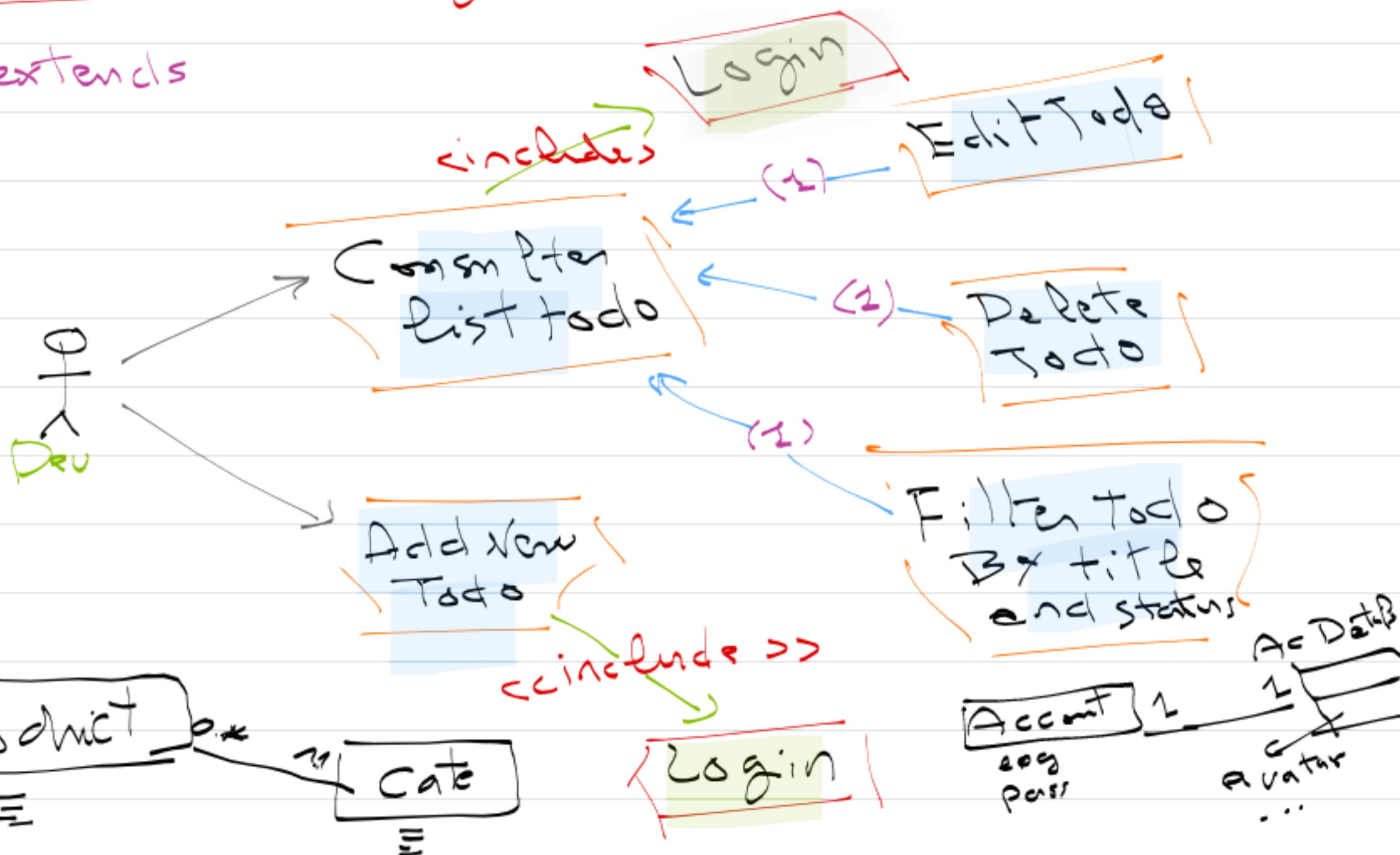
2) Diagramme de classe

→ identifier les données qui circule dans notre application

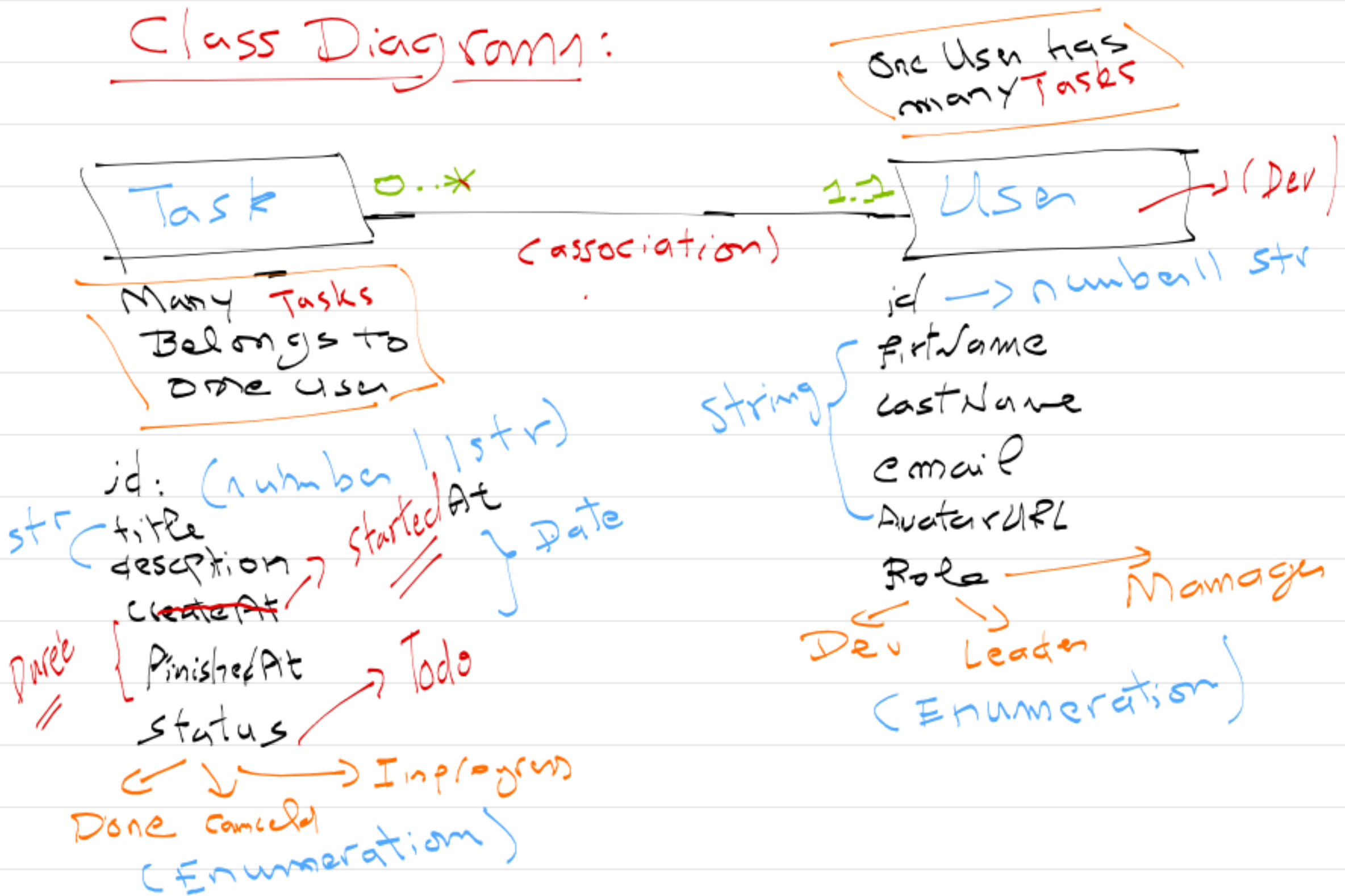
→ tâche (titre, desc, duration, status, creat At
→ user (first name, last name, role (U, D, M), email, ...)

Use Case Diagram:

1: extends



Class Diagram:

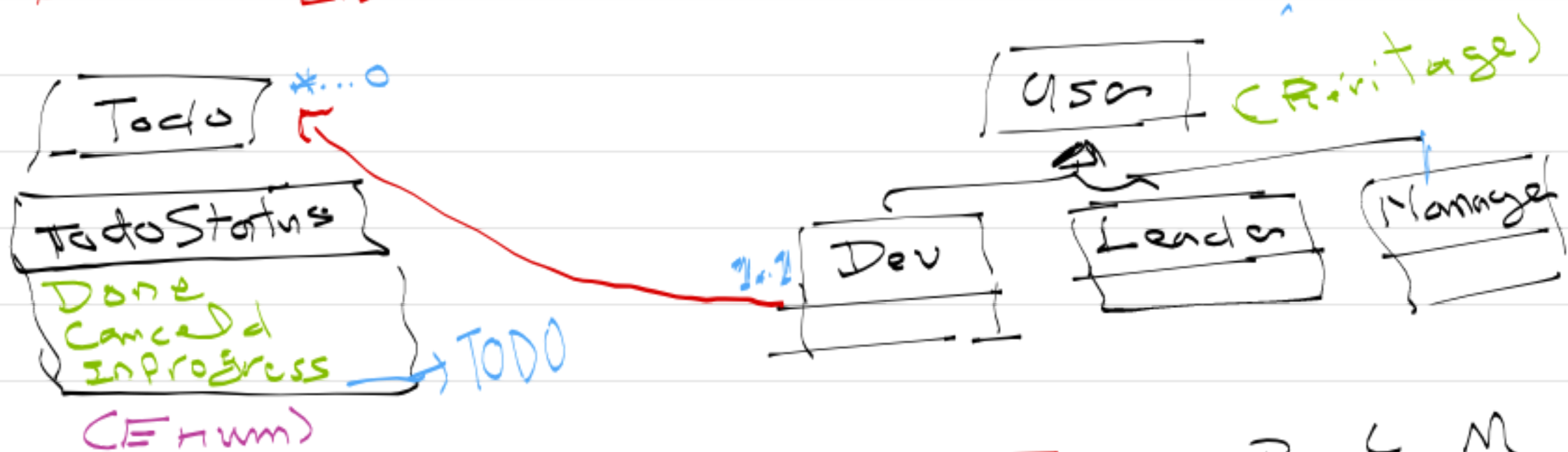


Backend

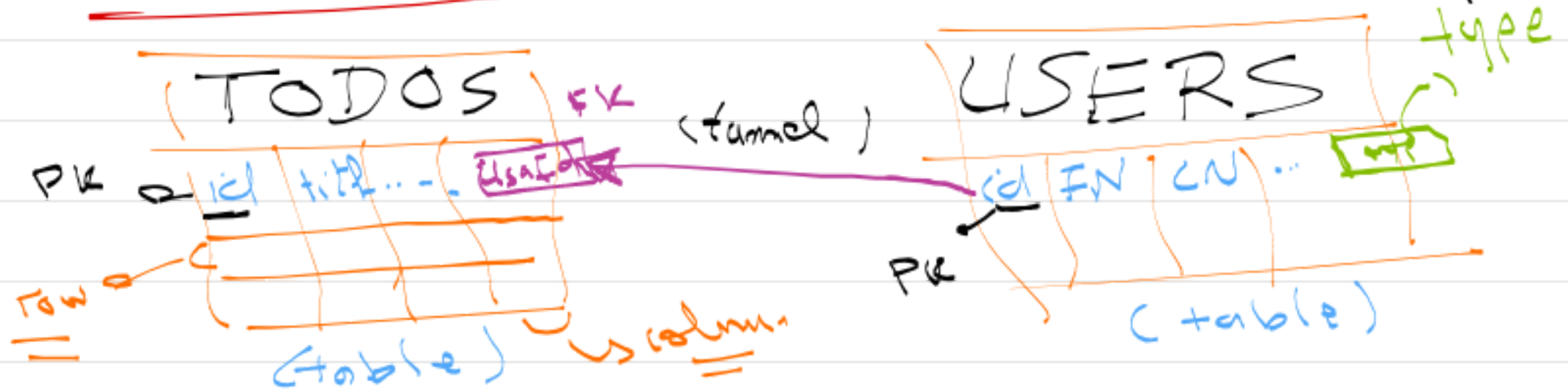
Data Base From Class Diagram:

- 1) Install Xamp
- 2) Create table In mysql
- 1) Diagram de class
- 2) convert to data Base

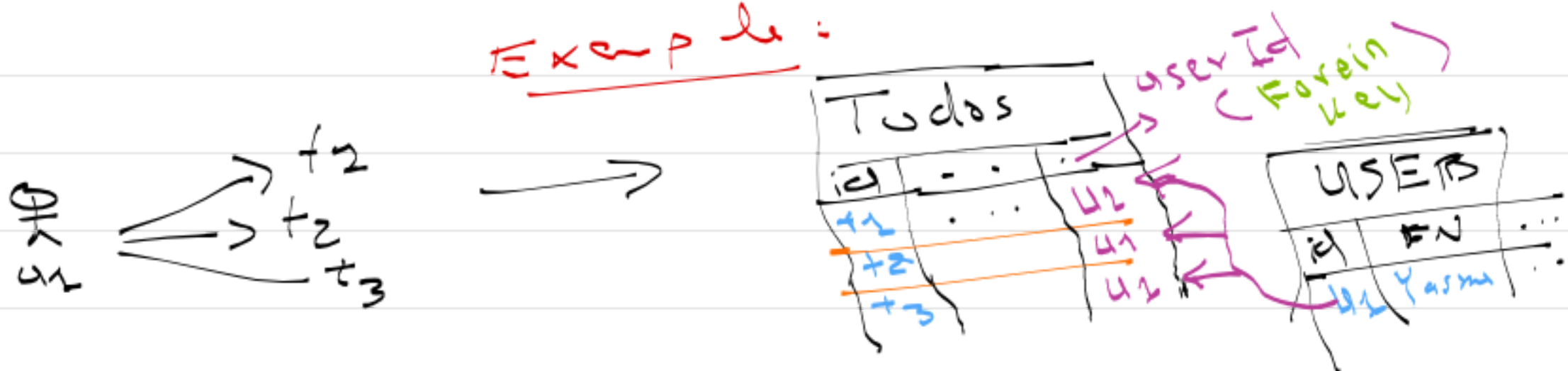
1) DC: todo List



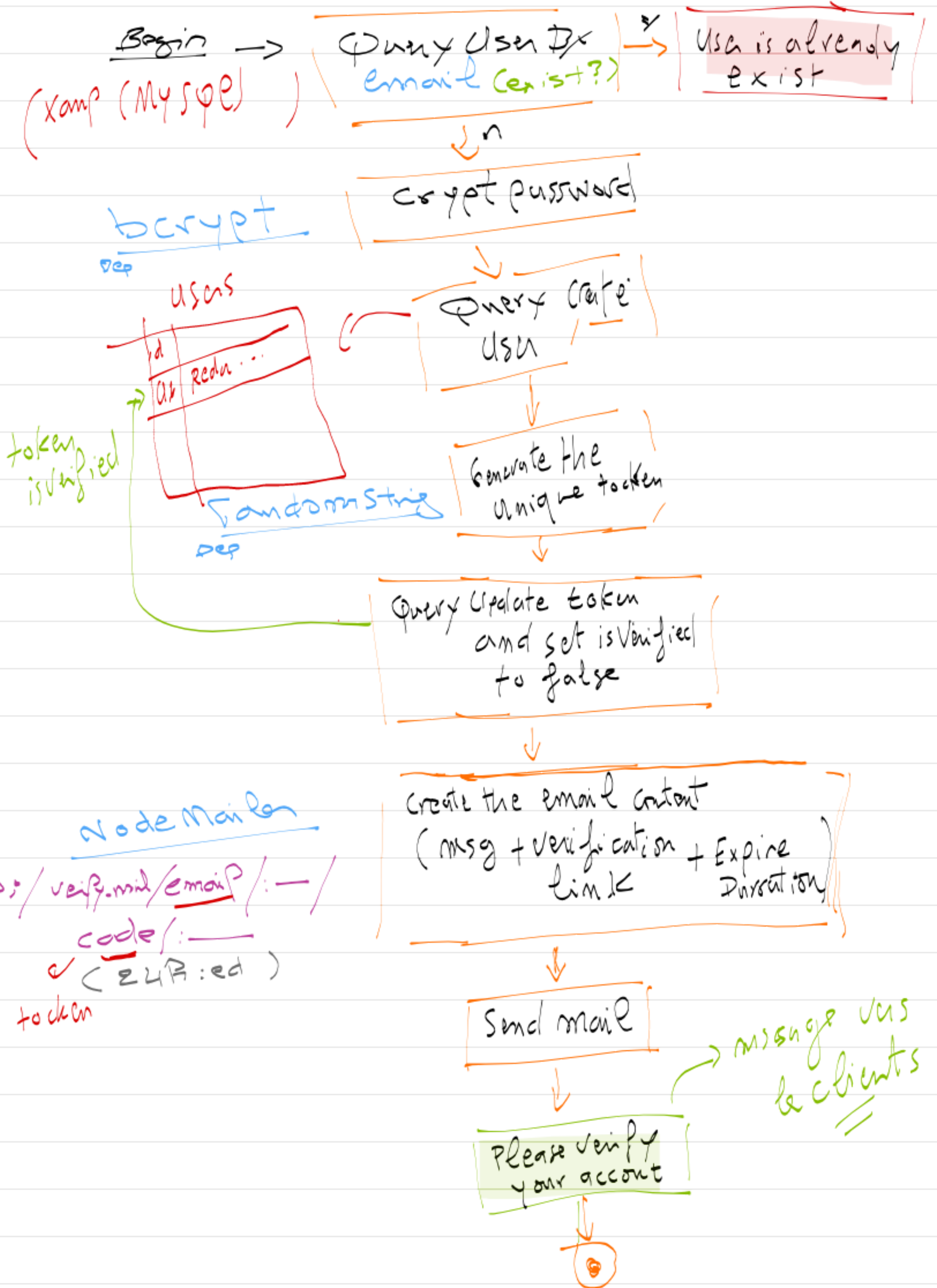
2) Convert to DP:



Example:



Flow charts (Inscription)



(runtime execution environment)

Backend

nodejs

Ecouter
les req
http

Communiquer
avec MySQL
(SPP)

Envoyer des
Reponses
http

Express

MySQL
(Driver)

Dependence
spéciale

node_modules

npm i -g node_modules

texte à coder
à exécuter
par le serveur

1)  node-Back

2) npm init

3) generate

Entry Point

package.json

4)  index.js