1. Capture of functional needs
2. Customer specifications

* Description de l'application :

Mise en place d'un système de gestion des documents (GED)

* Principales fonctionnalités :
* Gestion des documents
* Gestion des courriers entrants et sortants
* Mise en place d'un workflow de gestion des documents
* Utilisateurs
* Responsable bureau d'ordres
* Direction (RH, Financière, Comptabilité, ..)
* Administrateur
* Besoins fonctionnels :

Responsable bureaux d'ordre :

* S'authentifier et Scanner les documents à partir du système.
* Référencement des documents
* Gèrer un workflow
* Gèrer les courriers

Directeur :

* S'authentifier et gèrer des workflows.
* Gérer ses courriers
* Gestion des documents ( modification en ligne , téléchargement, modification et ajout des propriétés... ).
* Accéder aux statistiques.

Administrateur :

* Authentification.
* Gestion des utilisateurs.
* Gestion des workflows.
* Gestion des courriers.
* Accès aux statistiques

1. Preliminary Analysis
2. Analysis and critique of the existing solution

After analysing the current solution we discovered that our DSI department is still using the traditional way of archiving ,stocking document and document exchanging and we know that nowadays electronic documents archiving is dominant because it’s faster , safer and more garantied meanwhile the traditional way of archiving documents is no longer used by modern societies due to it’s drawbacks such as loss of time , probability to loose the document , no traçabilitity and other security issues.

1. Issues

The existing methode of archiving and handeling documents isn’t sufficient enough so we need to implement and application to make archiving and stocking documents electronic.

1. Our solution

We propose to develop an application to archive the documents and manage the workflow in the DSI .

1. Internal specifications
2. Functional requirements
3. Non-Functional requirements

* Security :The application has to offer full security for the client and consultant to manage any information which means accessible and usable only in authorized ways by authorized users.
* Performance : This one shouldn’t come as a surprise. Quality software has to be fast. Or at least feel fast. “It’s not fast enough” is a battle we never want to get into. When we think about an application being performant, we must think about specifying the following:

 Response times

How long should my app take to load? What about screen refresh times or choreography?

 Processing times

How long is acceptable to perform key functions or export / import data?

 Query and Reporting times :This could be covered off with general reporting times, in case of providing an API we should probably consider acceptable query times too.

* Capacity and Scalability : How much do we need to cope with now, and how much do you think we’ll need to cope with in the future. Unless you are truly blessed with an incredibly forgiving audience (or a great illustrator), you won’t be able to get away with your own fail whale.
* Availability : Our application must be available, users can edit or send information at any time.

 Hours of operation : When does our application need to be available: If we need to do a database upgrade or a system backup, can we take the system offline while you do it?

 Locations of operation : A few things to think about here: Geographic location, connection requirements and the restrictions of a local network prevail.

* Reliability :Our application has to be 100% reliable, clients should rely on it social networking services.
* Usability : Our application has to be easy to use, with standard looking interfaces and with clear and beautiful visual style.
* Maintainability :Our application must maintainable.