



12, rue de la Houssinière
44322 Nantes

Projet multi-modules

Getting Things Done

Rapport d'IHM

Louis-Alexandre Celton
Gaetan Hervouet
Damien Levin
Paul Vaillant
2009-2010

MASTER 2 - ALMA

Table des matières

1	Introduction	3
2	Présentation de l'architecture	4
3	Etape 1 : Développement du concept de produit	5
3.1	Objectifs du projet	5
3.2	Contraintes de l'IHM	5
3.3	Utilisateurs	6
3.4	Fonctionnalités principales	6
4	Etape 2 : Analyse	7
4.1	Segmentation des utilisateurs	7
4.2	Principales fonctionnalités	7
4.3	Scénarios de l'IHM	7
4.3.1	Identification	8
4.3.2	Collecte des informations	8
4.3.3	Organisation des tâches	9
4.3.4	Affichage des tâches	11
4.3.5	Réactualisation des tâches	11
4.3.6	Exigences fonctionnelles	12
4.3.7	Traitement des informations	12
4.3.8	Synchronisation	13
5	Etape 3 : Conception initiale de l'interface	15
5.1	Modèle de navigation	15
5.2	Look and Field	15
5.3	Rapidité d'exécution	16

5.4	Maquette de l'IHM	16
5.4.1	Connexion à l'application	16
5.4.2	Collecte des informations	17
5.4.3	Création des tâches	17
5.4.4	Organisation des tâches	18
5.4.5	Mise à jour des tâches	19
5.4.6	Visualisation	20
6	Etape 4 : Développement incrémental	22
7	Etape 5 : Implémentation de l'application	23
7.1	Outils utilisés	23
7.2	Aide et documentation	23
7.3	Interface finale	23
8	Etape 6 : Evaluation externe	28
8.1	Diffusion d'une version beta	28
8.2	Bug reporting	28
9	Conclusion	29

Chapitre 1

Introduction

L'objectif de ce rapport est de présenter la conception de notre interface homme-machine. Elle s'inscrit dans le cadre du projet multi-module du master 2 alma. Ce livrable est structuré selon la méthode de conception LUCID (Logical User-Centered Interactive Design) et se décompose donc en 5 parties.

1. Développement du concept de produit
2. Analyse
3. Conception initiale de l'interface
4. Implémentation de l'application
5. Évaluation

Chapitre 2

Présentation de l'architecture

La partie qui nous concerne ici correspond à l'interface du client Web de l'architecture suivante :

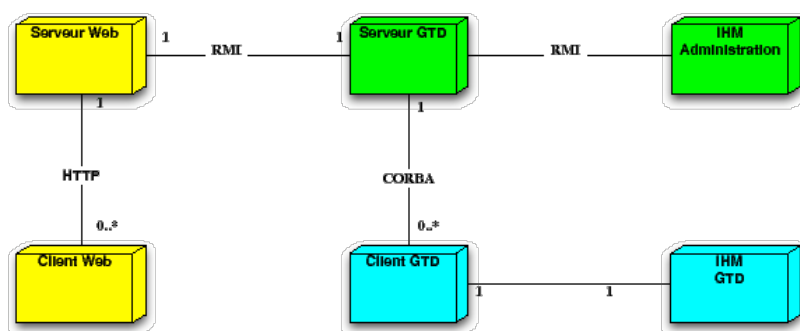


FIG. 2.1 – Architecture Générale

L'objectif de cette IHM est de gérer deux types de serveurs GTD (Getting things done) : un premier réalisé par une équipe du master2 alma, un second disponible gratuitement sur internet, nommé Toodledo¹.

¹<http://www.toodledo.com>

Chapitre 3

Etape 1 : Développement du concept de produit

3.1 Objectifs du projet

Le projet s'inscrit dans un cadre universitaire, ainsi il n'est pas destiné à être commercialisé. Cependant, ce projet est sous licence GPL3 et peut donc être distribué, et modifié librement. Aucune contrainte de conception n'est définie pour l'IHM, hormis le fait qu'elle doit respecter scrupuleusement la définition des fonctionnalités de la méthode GTD.

3.2 Contraintes de l'IHM

Le logiciel doit être utilisable facilement par n'importe quels types d'utilisateurs, il devra respecter les contraintes suivantes :

Facilité de prise en main, intuitivité : Le logiciel ne disposera pas de fonctions trop avancées pour ne pas déstabiliser l'utilisateur et pour gagner un maximum de temps, en allant à l'essentiel.

Aide en ligne : Le logiciel disposera de bulle d'aide (tooltips) pour aider l'utilisateur de façon ponctuelle. Une documentation papier ainsi qu'une formation seront également mis à disposition de l'utilisateur.

Textes en anglais et français : Internationalisation pour élargir le panel d'utilisateurs. Dans un premier temps, seul l'anglais et le français seront disponibles.

Respect de la méthode GTD : Le cahier des charges de l'application a été respecté. Ceci est détaillé dans le livrable de l'analyse du projet.

Respect des bonnes pratiques de conception des IHM Le logiciel respectera les conventions de base de toute Interface Homme Machine (Taille des boutons, disposition des menu en haut, lecture de l'IHM de gauche à droite...)

Rapidité de fonctionnement : L'IHM sera réactive. Cependant en cas de traitements longs, l'utilisateur sera averti du temps restant. Cela permet de faire patienter l'utilisateur, sans lui laisser croire que l'application ne réponds plus.

3.3 Utilisateurs

Le produit s'adresse à tous les types d'utilisateurs disposant des compétences de bases en informatique. Ainsi, ceux-ci doivent être en mesure de savoir naviguer sur internet et de savoir manipuler un logiciel relativement simple.

3.4 Fonctionnalités principales

L'objectif du projet est de fournir un logiciel complet de gestion de tâches et s'appuyant sur la méthode GTD¹. Le logiciel est multi-utilisateurs et est en mesure de fonctionner grâce à deux serveurs :

- Un serveur GTD créé par un groupe du master alma
- Un serveur ToodleDo

Il devra donc permettre la création de tâches, de projets, de contextes, etc... De plus, il devra permettre d'effectuer une synchronisation des données entre les deux serveurs, dans le cas où les deux serveurs ne seraient pas synchronisés. Ce genre de scénarios se produit par exemple lorsqu'un serveur est indisponible lors d'une mise à jour.

¹http://fr.wikipedia.org/wiki/Getting_Things_Done

Chapitre 4

Etape 2 : Analyse

4.1 Segmentation des utilisateurs

Comme nous l'avons dit, le logiciel sera utilisable par n'importe quels types d'utilisateurs. C'est pourquoi il n'y aura pas différentes classes d'utilisateurs pour l' IHM. Cependant pour le reste du système (Serveur GTD notamment) une classe administrateur est définie. Elle dispose des droits maximums sur le système et offre la possibilité d'administrer le serveur. Cette classe ne fait pas partie de notre projet (seulement sur le serveur GTD) et n'est donc pas décrite dans la suite du document.

4.2 Principales fonctionnalités

Les fonctionnalités de l'IHM respectent les principes de la méthode GTD. Ainsi, on retrouve les activités principales suivantes :

- Collecte des informations : Création de notes (idées) qui vont aboutir à la création de tâches
- Création des tâches
- Organisation des tâches : Manipulation des tâches et des projets
- Mise à jour des tâches
- Visualisation : Affichage des tâches selon le contexte

4.3 Scénarios de l'IHM

Voici une liste exhaustive des fonctionnalités offertes par le logiciel. L'ensemble des scénarios s'appuient sur des cas d'utilisations cockburn.

4.3.1 Identification

Use Case: 1 – Identification

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM, Base de données

Pré-condition : L'utilisateur est non connecté et déjà inscrit.

Post-condition : L'utilisateur est connecté.

Priorité : (5/5) : Cette étape est nécessaire pour chacun des prochains cas d'utilisation.

Fréquence : Chaque lancement d'application.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur entre son identifiant et son mot de passe,
- 2 Il valide,
- 3 L'utilisateur est connecté.

EXTENSIONS

- 1 Renvoyer le mot de passe
- 2 Création d'un compte

Exigences fonctionnelles

FONC11 - Saisie des identifiants,

FONC12 - Connexion (avec gestion de session),

FONC13 - Inscription,

FONC14 - Régénération de mot de passe.

4.3.2 Collecte des informations

L'application n'a normalement pas à intervenir dans cette étape. Le recensement des idées est effectivement un processus utilisateur. Cependant, un pense-bête à idées non traitées par l'utilisateur peut s'avérer fort utile dans le cas où l'utilisateur est interrompu dans son processus. Un pense-bête permet alors de stocker les idées non traitées.

Use Case: 2 – Collect

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM, Base de donnée

Partie prenante et intérêt : Le recensement exhaustif de tout ce qui peut justifier une quelconque intervention de notre part : en suspens, inachèvement, en attente, intention, projet, manque, usure, mauvais fonctionnement, problème, insatisfaction, besoin, engagements à tenir, etc. Exemples : cette carte de visite restée dans une poche, cette facture dans la boîte à gants, cette demande reçue, ce dossier qui traîne sur le bureau, cette agrafeuse qui coince, cette course à faire, ce problème à résoudre, cette suggestion à tester, les messages de la boîte vocale, ce projet jamais réalisé, ce souci de santé, ce fauteuil qui grince, les performances de ce collaborateur, toutes ces choses en retard...

Pré-condition : Aucune.

Post-condition : Les informations sont enregistrées dans l'application.

Priorité : (1/5) : Cette étape n'est pas indispensable, et peut être effectuée sans l'aide du système informatique.

Fréquence : Chaque début de journée.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur après avoir effectué sa réflexion, saisie les informations qu'il a recensées dans la boîte à idée de l'application.

Exigences fonctionnelles

FONC21 - Saisie des idées,

FONC22 - Stockage des idées,

FONC23 - Suppression des idées.

4.3.3 Organisation des tâches

Use Case: 3 – Organize

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM, base de données

Pré-condition : On dispose d'une liste de tâches issues de l'organisation des données.

Post-condition : Les tâches appartiennent ou non à un projet, sont organisées en séquence ou non.

Priorité : Haute

Fréquence : Après chaque traitement des données.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur regroupe les tâches par projet en fonction des propriétés communes de leurs contextes et de leurs intérêts communs.
- 1 Il peut aussi séquencer les tâches d'un projet.
- 2 Il a aussi la possibilité de créer des sous-projets afin de donner un aspect hiérarchique au projet principal.

EXTENSIONS

- 1 Si une tâche est isolée, elle n'appartient à aucun projet et n'est pas séquençable.
-

Exigences fonctionnelles

Pour regrouper les tâches l'utilisateur doit pouvoir :

FONC31 - créer/modifier/supprimer des projets,

FONC33 - affecter ou non des tâches à un projet,

FONC35 - affecter ou non des projets à un projet (sous-projets).

Exigences non-fonctionnelles

FONC36 - Les tâches et sous-projets d'un même projet doivent participer à un même but.

FONC37 - Les tâches d'une même séquence doivent appartenir au même projet.

FONC38 - Les tâches et sous-projets d'un même projet doivent avoir des propriétés communes dans leurs contextes respectif.

4.3.4 Affichage des tâches

Use Case: 4 – Affichage des tâches

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM

Pré-condition : On dispose d'une liste de tâches, de contextes, de projets...

Post-condition : Les tâches sont affichées selon les critères et vues choisies par l'utilisateur.

Priorité : Haute

Fréquence : Selon les besoins de l'utilisateur.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur affiche les tâches qu'il effectuera selon les projets qu'il a choisis et le contexte courant.
- 2 Les tâches sont présentées de différentes manières (échancier, agenda...).

Exigences fonctionnelles

L'utilisateur doit pouvoir via l'IHM :

FONC71- choisir les projets à réaliser.

FONC72- sélectionner la ou les vues dans lesquelles seront affichées les tâches (échancier, agenda...).

4.3.5 Réactualisation des tâches

Use Case: 5 – Réactualisation des tâches

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM, base de données

Pré-condition : On dispose d'une liste de tâches.

Post-condition : On délègue des tâches, celles finies sont supprimées, les vues sont mises à jour suivant le nouveau contexte, création automatique des tâches périodiques.

Priorité : Haute

Fréquence : Au moins une fois par jour.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur change le contexte courant.
- 2 Indique les tâches effectuées.
- 3 Délègue des tâches.

EXTENSIONS

- 2b L'utilisateur peut annuler ses modifications.

4.3.6 Exigences fonctionnelles

L'utilisateur doit pouvoir :

FONC41 - mettre à jour le contexte.

FONC42 - indiquer les tâches réalisées.

FONC43 - déléguer une tâche.

FONC44 - revenir en arrière.

4.3.7 Traitement des informations

Use Case: 6 – Process

CHARACTERISTIC INFORMATION

Acteur : Utilisateur

Niveau : Tâche principale

Portée : IHM, Base de donnée

Pré-condition : L'utilisateur est connecté au serveur web.

Post-condition : Une tâche est créée sur les serveur GTD et Toodledo.

Priorité : (5/5) : Cette étape est nécessaire pour chacun des prochains cas d'utilisation.

Fréquence : De façon ponctuelle.

MAIN SUCCESS SCENARIO

- 1 L'utilisateur regarde les éléments affichés dans sa boîte à idées.
- 2 L'utilisateur saisie les informations relatives à la tâche qu'il souhaite créer (nom, date debut, date fin, priorité, temps et energie requis ...).
- 3 L'utilisateur crée la tâche.

EXTENSIONS

- 1 L'utilisateur oublie de remplir un champ obligatoire.

Exigences fonctionnelles

- FONC51** - Affichage de la boîte à idées,
- FONC52** - Saisie des informations relatives à une tâche,
- FONC53** - Création d'une tâche,

4.3.8 Synchronisation

Use Case: 7 – Synchronisation

CHARACTERISTIC INFORMATION

Acteur : Serveur Web

Niveau : Tâche principale

Portée : Serveur GTD et Toodledo et Serveur Web

Pré-condition : Le serveur web n'est pas synchronisé avec le serveur GTD ni Toodledo

Post-condition : Le serveur web et les serveurs GTD et Toodledo sont synchronisés

Priorité : (5/5) : Cette étape est nécessaire pour assurer une persistance des informations.

Fréquence : A chacune des transactions

MAIN SUCCESS SCENARIO

- 1 L'utilisateur effectue une opération sur le client web
- 2 Le serveur web enregistre les modifications utilisateur en mettant à jour sa base de données

3 Le serveur web envoie les modifications effectuées aux serveurs

EXTENSIONS

- 1 La connexion entre les serveurs web et GTD et/ou ToodleDo est interrompue
- 2 La connexion entre les serveurs web et GTD et/ou ToodleDo se rétablit

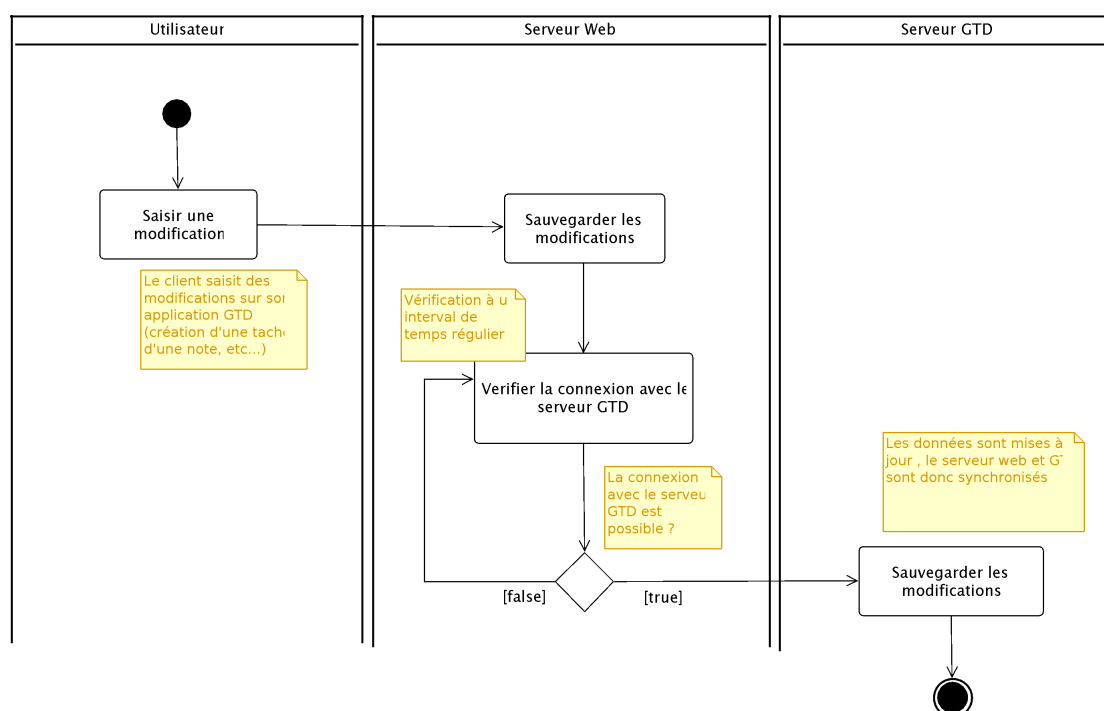


FIG. 4.1 – UML - Diagramme d'activités - Synchronisation

Exigences fonctionnelles

FONC61 - Enregistrement des modifications sur le serveur web,

FONC62 - Enregistrement des modifications sur les serveurs GTD et ToodleDo

FONC63 - Enregistrement des modifications sur les serveurs GTD et ToodleDo après une perte de connexion

Chapitre 5

Etape 3 : Conception initiale de l'interface

5.1 Modèle de navigation

L'application doit fournir les fonctionnalités de la méthode GTD qui définit 5 grandes étapes. L'interface graphique a donc été construite autour de celles-ci. En effet, au lancement de l'application, on remarque un axe principal où les 5 étapes apparaissent clairement. Des flèches rappellent l'ordre chronologique dans lequel elles sont sensées être effectuées. L'utilisateur reste cependant libre de cliquer sur l'étape qu'il souhaite effectuer à tout moment.

5.2 Look and Field

L'apparence de l'application est relativement sobre. Ainsi, le nombre de couleurs est limité afin d'offrir un rendu clair et fonctionnel. L'ensemble des éléments graphiques reste simple mais travaillé. Étant donné que nous utilisons GWT, les boutons, les tableaux, et les autres éléments ont tous le même style graphique défini par défaut dans GWT. Le thème général est cependant en adéquation avec ce thème par défaut utilisé par les boutons. Ceux-ci sont correctement placés dans l'interface de façon à ce que l'utilisateur soit efficace dans ses opérations. Les éléments cliquables sont signalés, tout comme chaque élément autorisant une interaction avec l'utilisateur.

L'application réalisée est de type RIA (Rich Internet Application). Elle possède donc les caractéristiques similaires aux logiciels traditionnels installés sur un ordinateur. L'interactivité est aussi développée que dans une application de bureau et la vitesse d'exécution est particulièrement rapide grâce au framework GWT qui optimise le code javascript de l'application cliente.

5.3 Rapidité d'exécution

Etant donné que l'application est réalisée en GWT, elle offre une réactivité particulièrement intéressante. Le logiciel se comporte comme une application de bureau standard. Lors des actions nécessitant un temps de réalisation plus long, l'utilisateur est cependant informé du temps restant via une barre de progression.

5.4 Maquette de l'IHM

Afin de faciliter le développement de l'IHM, les maquettes suivantes ont été réalisées. Elles s'organisent autour des principales étapes de la méthode GTD.

5.4.1 Connexion à l'application

Au lancement de l'application l'utilisateur doit pouvoir effectuer toutes les étapes de la méthode GTD. Cependant, la connexion est indispensable à chacune des étapes suivantes. En effet, les données sont stockées dans un compte distant identifié par un login et protégé par un mot de passe. L'interface suivante permet donc de se connecter, s'inscrire et renvoyer son mot de passe en cas de perte de celui-ci.



The mockup shows a web interface for user connection. At the top, there is a horizontal navigation bar with six buttons: 'Connect' (light gray), 'Collect' (dark blue), 'Process' (dark blue), 'Organize' (dark blue), 'Review' (dark blue), and 'View' (dark blue). Below this bar, the main content area has a light gray background. It contains two input fields: the first is labeled 'Identifiant' with the placeholder text 'Identifiant de connexion' below it; the second is labeled 'Mot de passe' with the placeholder text 'Mot de passe associé à votre compte' below it. At the bottom left, there are two buttons: 'Mot de passe oublié' and 'Inscription', both in light gray. At the bottom right, there is a dark blue button labeled 'Connexion'.

FIG. 5.1 – Mockup - Connection

Une fois que l'utilisateur est connecté, il peut effectuer toutes les étapes de la méthode GTD, ou seulement celle qui l'intéresse. L'utilisateur doit en effet être libre de choisir ce qu'il veut faire au moment où il lance l'application. Il peut donc le faire par l'axe présent en haut de l'interface, présentant dans l'ordre chronologique les différentes étapes.

5.4.2 Collecte des informations

Après avoir cliqué sur l'onglet Collect, l'utilisateur dispose d'une interface offrant les fonctionnalités d'un pense-bête. En effet, Collect est une réflexion de l'utilisateur, la seule aide que pour fournir l'application est un stockage des idées n'ayant pas eu le temps d'être traité dans Process. Il peut donc ajouter (bouton +) et supprimer (bouton -) simplement des idées, ici sous forme de post-it.

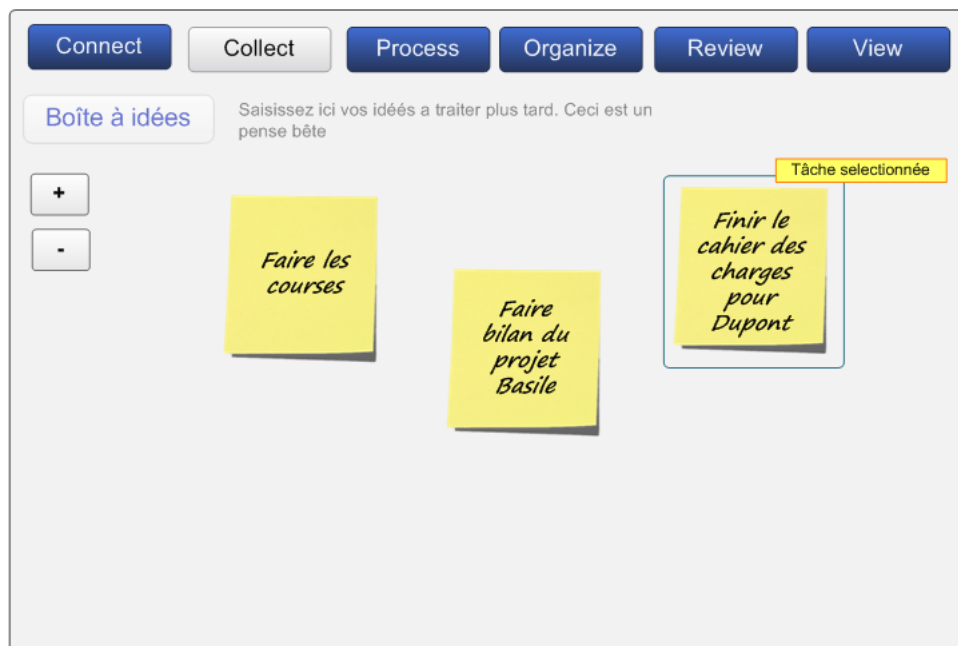


FIG. 5.2 – Mockup - Collect

5.4.3 Création des tâches

Lorsque l'utilisateur clique sur process, il dispose d'une interface affichant les idées qu'il a actuellement dans son pense-bête et d'une interface de saisie de tâches. Il peut donc aisément créer ses tâches avec toutes les informations nécessaires sans oublier celles qui sont dans son pense-bête.

FIG. 5.3 – Mockup - Process

5.4.4 Organisation des tâches

Lorsque l'utilisateur clique sur *organize*, il dispose de deux fonctionnalités :

- Organiser les tâches en projet,
- Déléguer une tâche.

L'organisation inclut la création de projet (bouton +), la définition du contexte par défaut du projet, ainsi que l'ajout des tâches dans un projet par glisser-déposer de la liste 'tâches disponibles' vers 'tâches affectées au projet', en définissant un ordre chronologique. Pour la délégation il suffit de sélectionner une tâche, puis une personne à qui déléguer, et enfin de valider.

The mockup shows a software interface for task management. At the top, there is a horizontal bar with six buttons: 'Connect', 'Collect', 'Process' (highlighted), 'Organize', 'Review', and 'View'. Below this bar, the 'Projets' section contains a dropdown menu set to 'Projet Basile' and a '+' button. A section titled 'Informations du projet' is separated by a horizontal line. It includes a 'Contexte par défaut' section with a text input field containing 'Lieu' and a '- State -' dropdown. Below this is an 'Outils' section with a list box and a '+' button. The main area is divided into two columns. The left column, titled 'Tâches disponibles', contains a list box with 'Faires les courses' and a scrollbar. The right column, titled 'Tâches affectées au projet', contains a list box with two items: '1-Finir le cahier des charges du projet Dupont' and '2-Faire le bilan du projet Basile', with a scrollbar. A section titled 'Délégation' is at the bottom, containing a 'Tâche' dropdown set to 'Faire les courses', a 'Deleguée à' dropdown set to 'Maman', and a 'Valider' button.

Connect Collect **Process** Organize Review View

Projets Projet Basile +

Informations du projet

Contexte par défaut Lieu - State -

Outils +

Tâches disponibles

Faires les courses ▼

Tâches affectées au projet

1-Finir le cahier des charges du projet Dupont
2-Faire le bilan du projet Basile

Délégation

Tâche Faire les courses ▼

Deleguée à Maman ▼

Valider

FIG. 5.4 – Mockup - Process

5.4.5 Mise à jour des tâches

L'interface Review permet simplement de mettre à jour les informations de certaines tâches. Elle correspond donc à l'interface de process.

Connect Collect Process Organize Review View

Projets Projet Basile

Informations du projet

Nom de la tâche

Titre succinct définissant au mieux la tâche

... ensemble des informations de la tâche ...

Periodicité Mensuel

- Journalière
- Hebdomadaire
- ✓ Mensuel
- Annuel

Modifier Annuler

FIG. 5.5 – Mockup - Review

5.4.6 Visualisation

La vue est l'opération finale, elle permet d'afficher les tâches en agenda ou en échéancier afin de conseiller l'utilisateur dans la bonne gestion de son temps. Le calendrier affiche les jours occupés et la liste des tâches présentes dans l'ordre de leur priorité depuis la date sélectionnée dans le calendrier.



FIG. 5.6 – Mockup - View

Chapitre 6

Etape 4 : Développement incrémental

Comme le montre la figure suivante, nous avons suivi un cycle de développement itératif ou incrémental pour la partie IHM. Chaque cycle est composé des trois étapes suivantes : Conception, Prototypage, Evaluation. Ce mode de développement permet d'avancer rapidement dans le projet en ayant toujours une base fonctionnelle et relativement stable. Il est ainsi facile de contrôler le bon avancement du projet.

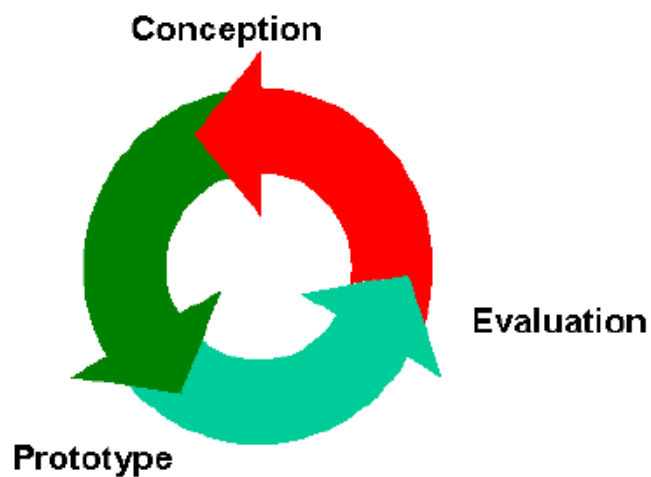


FIG. 6.1 – Développement incrémental

Chapitre 7

Etape 5 : Implémentation de l'application

7.1 Outils utilisés

L'IHM est réalisé grâce à la librairie GWT développée par google. Celle ci permet la création simple et rapide d'application AJAX. Elle permet de s'abstraire de la manipulation de javascript et de la gestion des appels asynchrones.

7.2 Aide et documentation

En ce qui concerne la documentation, le programme est livré avec une documentation papier. Celle-ci explique chaque fonction du logiciel en détail, tout en restant claire pour un utilisateur non expérimenté¹. De plus, l'IHM dispose de nombreuses bulles d'aide (tooltips) dans le but d'offrir une aide concise et rapide.

Pour ce qui est de l'installation de l'application et du passage de main de celle-ci, une brève formation sera réalisée. Elle sera destinée à l'utilisateur principal du logiciel, ou de son administrateur.

7.3 Interface finale

Voici des impressions écran de l'interface finale, celle-ci reprend l'ensemble des éléments décrit dans ce rapport.

¹Par manque de temps celle-ci n'a pu être finalisée

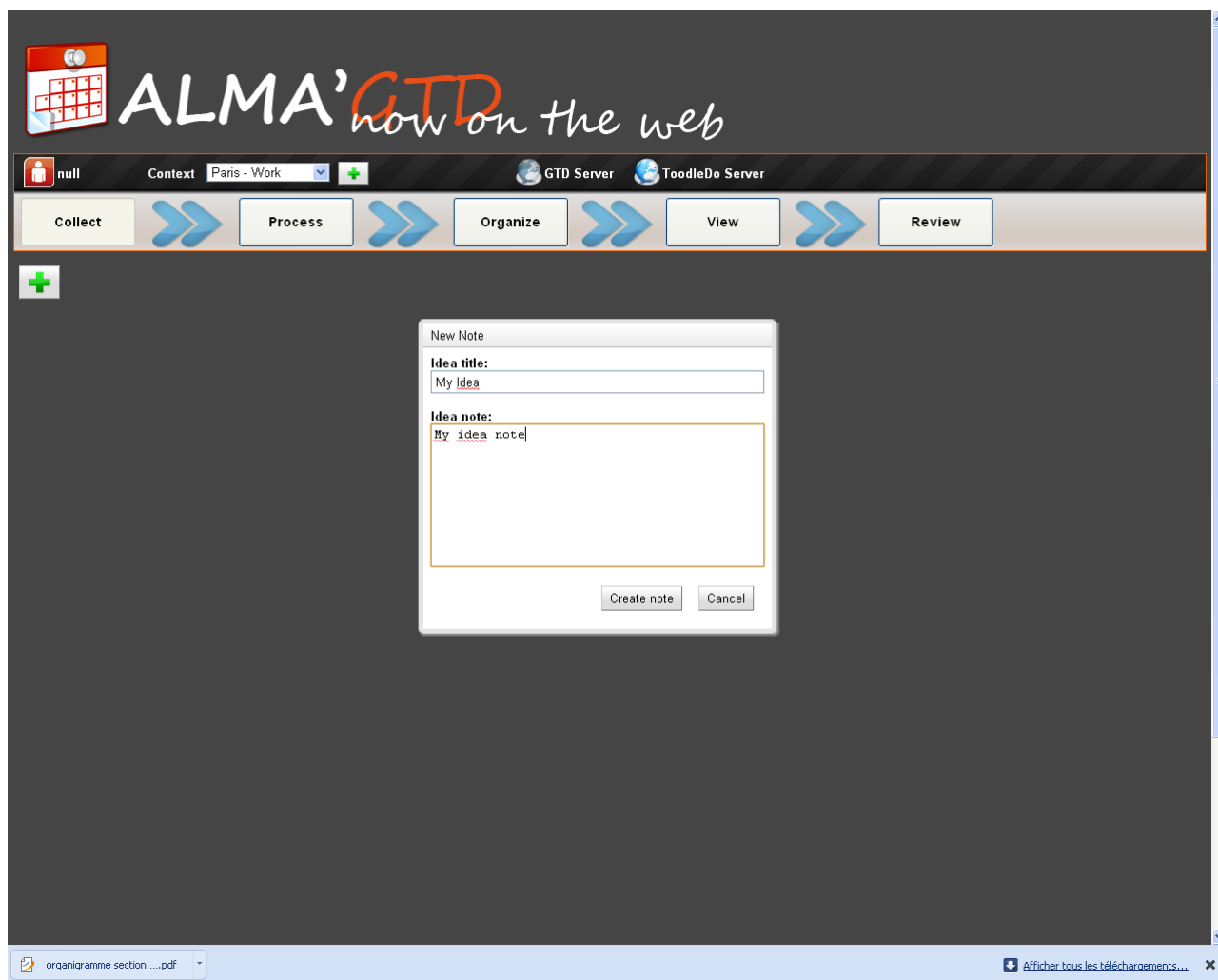


FIG. 7.1 – Collecte des informations



FIG. 7.2 – Collecte des informations

The screenshot displays the ALMA'GTD web interface. At the top, there's a header with the logo and the text "ALMA'GTD now on the web". Below this is a navigation bar with buttons for "Collect", "Process", "Organize", "View", and "Review". The main content area is divided into two sections: "IdeasBox" on the left and "Current Tasks List" on the right. The "IdeasBox" contains a list of ideas with "Load" buttons. The "Current Tasks List" contains a table of tasks with columns for Name, Effort Rate, Start Date, End Date, Frequency, Progress, ProjectId, and State. Below the table is a "Create" form with fields for Name, Effort Rate, Start Date, End Date, Frequency, Progress, ProjectId, and State, along with "Save", "Add", and "Del" buttons.

IdeasBox

- My Idea
- ▼ Faire les courses
Du pain et du boursin :)

Current Tasks List

Name	Effort Rate	Start Date	End Date	Frequency	Progress	ProjectId	State
newTask0	00	01/01/2010	02/01/2010	Once	00		To Do
newTask1	10	01/01/2010	02/01/2010	Once	10		To Do
newTask2	20	01/01/2010	02/01/2010	Once	20		To Do
newTask3	30	01/01/2010	02/01/2010	Once	30		To Do
newTask4	40	01/01/2010	02/01/2010	Once	40		To Do
My Idea	0	01/01/2010	02/01/2010	Daily	0		To Do

Create

Name : Effort Rate :

Start Date : End Date :

Frequency : Progress :

ProjectId : State :

FIG. 7.3 – Création des tâches

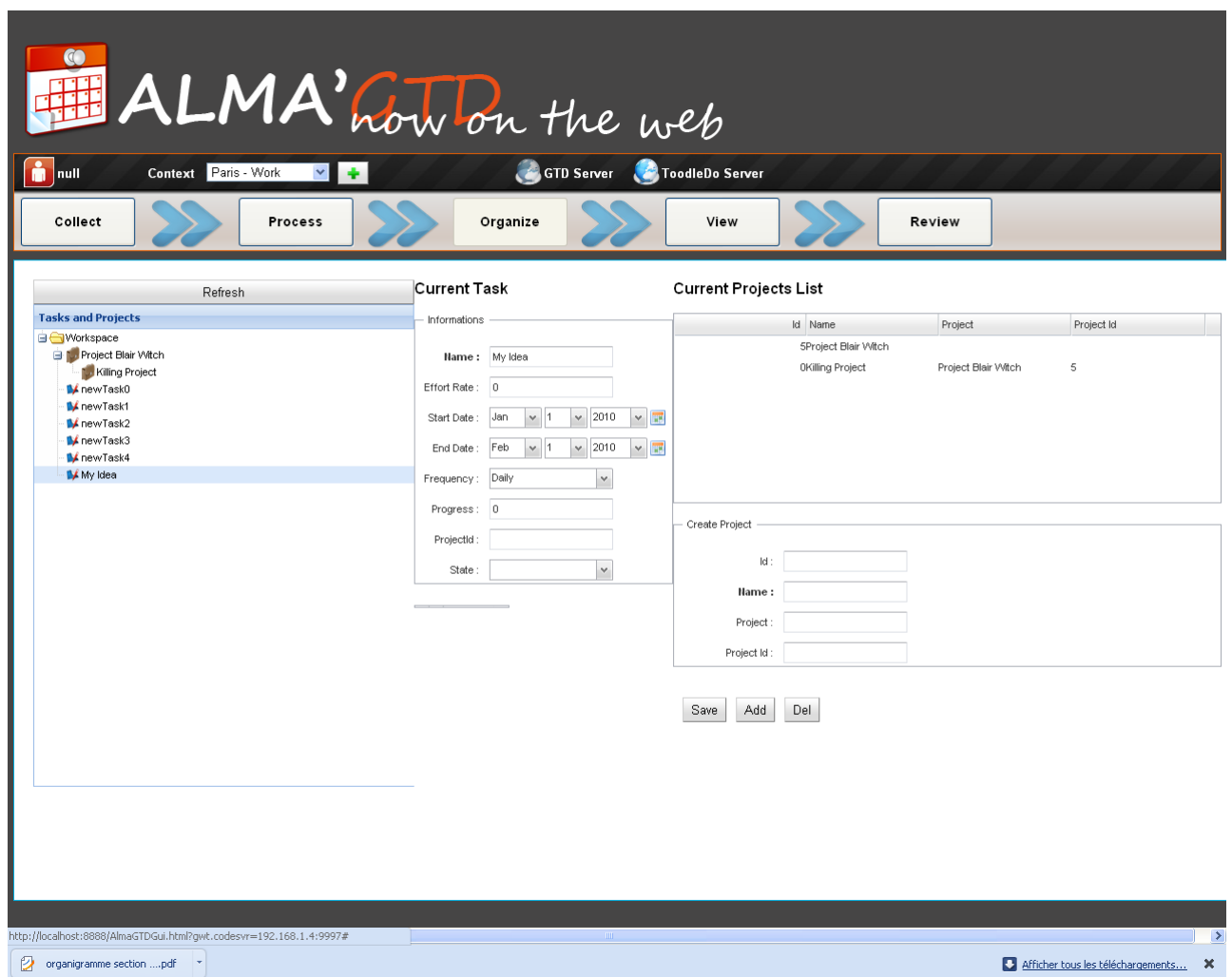


FIG. 7.4 – Organisation des tâches

Chapitre 8

Etape 6 : Evaluation externe

8.1 Diffusion d'une version beta

Dans le but d'aboutir à une version stable, et finalisée. Une première version de test sera mise à disposition des utilisateurs. Celle ci va permettre de recenser les bugs, et les problèmes d'utilisation. Cette étape est primordiale car elle permet d'avoir un avis externe à celui de l'équipe de développement. Les bugs seront tous corrigés avant la diffusion d'une autre version de test¹

8.2 Bug reporting

Le principal moyen de bug reporting se fera via internet et le site du projet². Toujours via ce site, les utilisateurs ont la possibilité de discuter avec l'équipe de développeurs et peuvent donc faire part de leurs avis.

¹Diffusion d'une RC1 et éventuellement d'une RC2

²[http ://code.google.com/p/almagtd/issues/list](http://code.google.com/p/almagtd/issues/list)

Chapitre 9

Conclusion

La réalisation de ce projet a été particulièrement intéressante pour nous. Bien que nous avons déjà eu des cours d'IHM par le passé, nous n'avons jamais utilisé GWT. Cette technologie c'est révélée très puissante pour le développement d'applications Web de par la compatibilité et l'optimisation qu'elle offre pour le client. Il n'est en effet plus nécessaire de manipuler du javascript et de se heurter aux nombreux problèmes de compatibilité avec les navigateurs, bêtes noires du développement Web actuel. De plus, GWT est un framework extrêmement efficace lorsqu'on la pris en main.

D'autre part, la réalisation de ce projet nous à permis de mettre en oeuvre tous les principes de développement d'IHM étudiés en cours. Pour cela, nous avons utilisé la méthode de développement LUCID. Le fait d'avoir réalisé une application Web à également été très positif pour l'ensemble de notre groupe, plus souvent habitué à réaliser des applications en SWING. Grâce à la puissance de GWT, il est désormais possible pour une application Web, de rivaliser avec une application classique.

Table des figures

2.1	Architecture Générale	4
4.1	UML - Diagramme d'activités - Synchronisation	14
5.1	Mockup - Connection	16
5.2	Mockup - Collect	17
5.3	Mockup - Process	18
5.4	Mockup - Process	19
5.5	Mockup - Review	20
5.6	Mockup - View	21
6.1	Développement incrémental	22
7.1	Collecte des informations	24
7.2	Collecte des informations	25
7.3	Création des tâches	26
7.4	Organisation des tâches	27