

Modeleur de Formulaires

Document de Spécifications – Open Paas

Table des matières

1 Contexte.....	5
1.1 Description.....	5
1.2 Utilisateurs Cible.....	5
1.3 Conditions d'Utilisation.....	5
1.4 Besoins.....	6
2 Cas d'Usage.....	7
2.1 Modéliser un Formulaire.....	7
2.2 Gérer des Modèles de Formulaires.....	7
2.3 Instancier un Modèle de Formulaire.....	7
2.4 Gérer des Instances de Formulaires.....	8
2.5 Remplir une Instance de Formulaire.....	8
2.6 Récupérer les Résultats d'une Instance de Formulaire.....	9
2.7 Enchaînement des Cas d'Usages.....	9
3 Tâches.....	10
3.1 Concepts.....	10
3.2 Tâches liées à un Modèle de Formulaire.....	10
3.2.1 Créer un Modèle de Formulaire.....	10
3.2.2 Lister les Modèles de Formulaires.....	11
3.2.3 Supprimer un Modèle de Formulaire.....	11
3.2.4 Éditer un Modèle de Formulaire.....	11
3.3 Tâches liées à une Instance de Formulaire.....	12
3.3.1 Créer une Instance de Formulaire.....	12
3.3.2 Lister les Instances de Formulaires.....	12
3.3.3 Supprimer une Instance de Formulaire.....	13
3.3.4 Éditer une Instance de Formulaire.....	13
3.3.5 Remplir une Instance de Formulaire.....	13
3.3.6 Visualiser les Résultats d'une Instance de Formulaire.....	14
3.4 Tableau Associatif entre cas d'usage et tâches.....	14
4 IHM Abstraite.....	15
4.1 Charte Graphique.....	15
4.2 Maquettes pour la liste des Modèles.....	15
4.3 Maquettes pour les Instances de Formulaires.....	18
4.4 Maquettes pour la Modélisation d'un Formulaire.....	18
5 Contraintes Techniques et Implémentation.....	19
5.1 Contraintes Techniques.....	19
5.2 Gestion des Erreurs.....	19

Ce document est la spécification du modeleur de formulaires.

Cet outil est une brique du projet Open Paas. Il supplante ce qui devait à l'origine être un moteur de workflow social implémenté en BPEL. Les raisons de ce changement ont été expliquées dans le rapport d'avancement de T0 + 20.

Il est à noter que l'intégration dans le reste de la plateforme Open Paas est traitée séparément de cette spécification. C'est notamment le cas de la vérification des profils et donc du contrôle d'accès aux différentes parties de l'application.

1 Contexte

1.1 Description

Le modeleur de formulaires est une application web qui permet de modéliser des formulaires web. L'idée est de pouvoir construire graphiquement de tels formulaires, et de pouvoir ensuite les déployer dans la plateforme Open Paas.

Cet outil constitue donc une façon de créer des applications SaaS simples.

Elles couvrent un certain nombre d'utilisations, de la création de sondages jusqu'à des formulaires administratifs.

1.2 Utilisateurs Cible

On différencie 3 types d'utilisateurs :

- Le **concepteur** est une personne qui va créer un squelette / modèle de formulaire. C'est lui qui va assembler des éléments graphiques et les organiser pour présenter un formulaire complet. C'est lui aussi qui définit les règles de validation du formulaire.
- Le **responsable de publication** est la personne qui va instancier le modèle de formulaire, lui associer un contexte et rendre le formulaire accessible à des gens au travers d'une adresse internet. Il est également capable de récupérer les résultats du formulaire publié.
- Le **sondé** est une personne qui va venir remplir un formulaire, une fois celui-ci publié.

Il existe 2 profils de concepteur.

Un concepteur peut être un développeur qui souhaite créer un formulaire rapidement, et qui souhaite gagner du temps. Pour lui, la sortie du modeleur est une page web qu'il pourra retoucher à la main.

Enfin, et c'est le cas le plus large, un concepteur peut être une personne sans compétence particulière en matière de programmation. C'est quelqu'un qui a besoin d'un outil informatique simple pour interagir dans son métier avec d'autres personnes. Un exemple de ce profil est un responsable administratif dont les connaissances informatiques se limitent aux outils de bureautique.

1.3 Conditions d'Utilisation

Cette application s'accède comme un site web.

Elle a vocation à s'intégrer dans Open Paas, mais peut aussi être vue comme une application indépendante.

Elle est utilisée au travers d'un navigateur Web pouvant s'exécuter sur un ordinateur fonctionnant sous Windows, Linux ou Mac OS, ou bien encore depuis une tablette. Seule la restitution des résultats est accessible depuis les navigateurs mobiles (smartphones).

1.4 Besoins

Voici une liste haut-niveau des besoins à couvrir par cette application. Ils sont déclinés en cas d'usage dans la section suivante.

- Modéliser un formulaire.
- Gérer des modèles de formulaires.
- Instancier un modèle de formulaire.
- Gérer des instances de formulaires.
- Remplir une instance de formulaire.
- Récupérer les résultats d'une instance de formulaire.

Il faut noter aussi que le mot **besoin** peut être remplacé par **fonctionnalité** ou **cas d'usage**. Les fonctionnalités sont destinées à tous les types d'utilisateurs. Simplement, une fonctionnalité peut avoir plus d'intérêt pour certains utilisateurs que pour d'autres.

2 Cas d'Usage

Un diagramme UML des cas d'usage n'a pas été jugé pertinent.

2.1 Modéliser un Formulaire

L'application doit permettre de modéliser une formulaire.

Cette modélisation se fait au travers d'une construction graphique, par glisser-déposer d'éléments pris dans une palette et déplacés vers une zone d'assemblage.

Le modèle définit une seule page, contenant des widgets ou composants graphiques (labels, boutons, listes déroulantes, case à cocher, champs de texte, champs pour un mot de passe, champs de recherche, boutons radio, boutons de validation).

Un bouton dans la page permet de sauvegarder son modèle sous un nom choisi par l'utilisateur. Un autre bouton permet de repartir à zéro et donc d'effacer le modèle affiché.

Un modèle de formulaire possède un titre, une description, une date de création et un auteur.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
x	Non	Non

2.2 Gérer des Modèles de Formulaires

L'application doit permettre de gérer les modèles créés et sauvegardés.

Elle doit donc être en mesure de lister tous les modèles sauvegardés. La liste doit afficher tous les champs d'informations d'un formulaire (titre, description, date de création et auteur).

Pour chaque item de cette liste, on doit pouvoir sélectionner un ou plusieurs modèles et pouvoir les supprimer.

En sélectionnant un modèle, on doit être en mesure de pouvoir l'éditer, c'est à dire de retourner à l'étape de modélisation.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
x	Non	Non

2.3 Instancier un Modèle de Formulaire

Un modèle de formulaire ne peut pas être rempli par une personne.

Pour que ce soit possible, il faut publier / instancier ce modèle. Une instance d'un formulaire est une copie du modèle, associée à une URL et accessible à des gens qui vont pouvoir remplir ce formulaire.

Pour instancier un modèle de formulaire, l'application doit lister les modèles disponibles et permettre de sélectionner le modèle à instancier. Il doit être possible de voir les informations du modèle (titre, description, date de création et auteur).

Une fois sélectionné dans la liste, le modèle est affiché, en donnant ainsi un aperçu. Un lien permet alors à l'utilisateur d'instancier ce modèle.

Une instance de formulaire a un titre, une description, une date de création, une date de début, une date de fin et un auteur. La date de début et la date de fin permettent de planifier la période durant laquelle cette instance sera accessible. Elle a aussi une URL, générée et basée notamment sur le titre. Toutes ces informations sont indépendantes du modèle. Une fois l'instanciation faite, l'instance conserve une référence vers le modèle, mais une modification faite sur le modèle n'impactera pas l'instance.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
Non	x	Non

2.4 Gérer des Instances de Formulaires

L'application doit permettre de gérer l'ensemble des instances de formulaires. Il faut donc pouvoir lister l'ensemble des instances et afficher leurs informations (URL d'accès, titre, description, date de création, date de début, date de fin, auteur, et le statut – ouvert ou fermé - déduit des dates de début et de fin). Il est possible de sélectionner une ou plusieurs instances et de les supprimer.

Une instance sélectionnée isolément dans la liste peut être éditée. Tous les champs (méta-données) peuvent être édités. La modification du titre impactera l'URL d'accès. Seul le contenu du formulaire ne peut pas être édité. Autrement dit, les widgets et les règles de validation ne peuvent pas être modifiés.

Si une instance est modifiée, un lien permet de sauvegarder les modifications.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
Non	x	Non

2.5 Remplir une Instance de Formulaire

Une fois qu'une instance a été créée, elle accessible au travers d'une URL.

Cette URL partagée, des utilisateurs vont pouvoir venir remplir cette instance (il s'agit de remplir un formulaire web classique).

Les règles de validation sont vérifiées au fur et à mesure que les champs sont remplis. En bas du formulaire, un bouton permet d'enregistrer les résultats.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
Non	Non	x

2.6 Récupérer les Résultats d'une Instance de Formulaire

A tout moment, un responsable de publication peut observer les résultats obtenus pour une instance de formulaire. Les résultats sont accessibles de diverses manières :

- Accès aux réponses individuelles, c'est à dire qu'une question = 1 réponse.
- Accès aux totaux, c'est à dire que pour une question donnée, on voit l'ensemble des réponses obtenues et lorsque c'est possible, la répartition des réponses.

Utilisateurs Concernés		
Concepteur	Responsable de Publication	Sondé
Non	x	Non

2.7 Enchaînement des Cas d'Usages

Les cas d'usages décrits précédemment forment en fait les activités successives d'un workflow faisant intervenir plusieurs utilisateurs. Cet enchaînement est synthétisé ci-dessous.

Activité	Acteur
1. Création d'un modèle	Concepteur
2. Instanciation d'un Modèle	Responsable de Publication
3. Remplissage d'une Instance	Sondé
4. Récupération des Résultats d'une Instance	Responsable de Publication

Les autres cas d'usages, non-listés dans ce tableau, correspondent à des transitions ou des possibilités supplémentaires offertes aux utilisateurs.

3 Tâches

Nous avons listé les besoins, c'est à dire, les fonctionnalités attendues par l'utilisateur mais vues sous le prisme « génie logiciel ». Nous allons maintenant traduire ces besoins en une suite de tâches, c'est à dire voir la réalisation de ces besoins sous le prisme « interface homme-machine ».

Rappels

Une tâche traduit un but utilisateur.

L'utilisation d'une fonctionnalité passe par l'exécution de tâches. Une tâche peut être composée de sous-tâches ou bien être élémentaire (pas de décomposition possible). Une composition de tâches peut se faire via une relation de dépendances ou bien via une relation temporelle.

Dans la pratique, une tâche commence par un verbe.

Ce verbe traduit une action utilisateur mais pas au niveau informatique : choisir, sélectionner, consulter, actionner...

Chaque tâche doit prévoir des scénarios : cas normal, cas d'exception, cas d'interruption. On peut également préciser pour chaque tâche les concepts manipulés (quels objets intéressent l'utilisateur), les pré et postconditions, des contraintes (durée maximale par exemple), la fréquence de cette tâche, sa difficulté et sa criticité.

Les sections suivantes décrivent un ensemble de tâches.

Chaque tâche est rattachée à un ou plusieurs besoins. Ces associations tâches – besoins sont résumées dans un tableau à la fin de cette section. L'identification des tâches est nécessaire pour définir une IHM abstraite.

3.1 Concepts

Les concepts sont les objets manipulés par les utilisateurs dans l'application.

Dans notre application, il y a 2 concepts essentiels :

- Un **modèle de formulaire** décrit le contenu graphique et les règles de validation d'un formulaire.
- Une **instance de formulaire** est un modèle de formulaire qui a été associé à une URL et qui est prêt à être rempli. Une instance possède des résultats.

3.2 Tâches liées à un Modèle de Formulaire

Les tâches représentent des actions utilisateurs dans l'application.

Une tâche peut dépendre d'autres tâches.

La notation suivante est prise :

- [tâche] fait référence à une autre tâche.
- Action (sans crochets) fait référence à une action utilisateur.

3.2.1 Créer un Modèle de Formulaire

Identifiant : MOD-1

Concepts associés : modèle de formulaire.

Préconditions : -

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Jean va sur la page de création de formulaire. Il entre un nom (« Qui est-tu ? ») et une description (« En apprendre plus sur les gens ») pour le formulaire qu'il va construire. Puis il fait glisser un label, auquel il donne la valeur « Nom : ». Il fait ensuite glisser un champs texte. Il précise pour ce champs texte qu'il ne peut pas être vide. L'utilisateur devra remplir ce champs. Enfin, il fait glisser un autre label (« Sexe: ») et une liste déroulante, à laquelle il donne les valeurs « Homme » et « Femme ». Jean enregistre son modèle.

Scénario d'erreur : pas de scénario d'erreur. Il est possible de créer des formulaires avec le même nom et la même description. La date de création fera la différence.

3.2.2 Lister les Modèles de Formulaires

Identifiant : MOD-2

Concepts associés : modèle de formulaire.

Préconditions : -

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Jean va sur la page listant les modèles de formulaires. La page liste l'ensemble des modèles déjà créés. Pour chaque modèle, on trouve le nom, la description, la date de création et son auteur.

Scénario d'erreur : Jean va sur la page listant les modèles de formulaires. Un message lui indique qu'aucun modèle n'a été créé. Un lien lui propose de créer un modèle.

3.2.3 Supprimer un Modèle de Formulaire

Identifiant : MOD-3

Concepts associés : modèle de formulaire.

Préconditions : [MOD-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Jean vient de [lister les modèles de formulaires]. Il en sélectionne 2 et demande à les supprimer. Une confirmation lui est demandée. Il confirme la suppression. Les modèles sont supprimés.

Scénario normal alternatif : Jean vient de [lister les modèles de formulaires]. Il en sélectionne 2 et demande à les supprimer. Une confirmation lui est demandée. Il a changé d'avis et invalide sa demande de suppression. Il retourne à la [liste des modèles de formulaires].

3.2.4 Éditer un Modèle de Formulaire

Identifiant : MOD-4

Concepts associés : modèle de formulaire.

Préconditions : [MOD-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Jean vient de [lister les modèles de formulaires]. Il sélectionne un modèle et demande à le modifier. Il retombe sur l'interface de [création d'un formulaire]. La seule différence est que les champs titre et description sont pré-remplis, et que la zone de construction contient le modèle déjà construit, prêt à être édité. Une fois les modifications faites, Jean demande à enregistrer la version retouchée.

Scénario normal alternatif : Jean vient de modifier un modèle de formulaire, mais décide finalement de ne pas enregistrer la modification. Il lui suffit de changer de page. Un bouton lui propose d'annuler l'édition et de revenir à la [liste des modèles de formulaires]. Il n'y a pas de sauvegarde automatique.

3.3 Tâches liées à une Instance de Formulaire

3.3.1 Créer une Instance de Formulaire

Identifiant : INST-1

Concepts associés : modèle de formulaire, instance de formulaire.

Préconditions : [MOD-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Paul [liste les modèles de formulaires]. Il en sélectionne un et demande à l'instancier. Une page lui demande de donner un nom, un titre, un date de début et une date de fin à l'instance. Une option lui permet d'ouvrir de démarrer l'instance immédiatement. Et une autre option lui permet de laisser l'instance ouverte indéfiniment (sans date de fin). Après que ces éléments aient été remplis, Paul enregistre l'instance.

Scénario normal alternatif : Après avoir complété tous les champs pour créer l'instance, Paul décide finalement qu'il ne veut pas publier d'instance de ce formulaire. Un bouton « annuler » lui permet de revenir à la [liste des modèles de formulaires].

3.3.2 Lister les Instances de Formulaires

Identifiant : INST-2

Concepts associés : instance de formulaire.

Préconditions : -

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Paul va sur la page listant les instances de formulaires. La page liste l'ensemble des instances déjà créées. Pour chaque instance, on trouve le nom, la description, son auteur et son statut (ouvert ou fermé - déduit des dates de début et de fin).

Scénario d'erreur : Paul va sur la page listant les instances de formulaires. Un message lui indique qu'aucune instance n'a été créée. Un lien lui propose de revenir à l'accueil.

3.3.3 Supprimer une Instance de Formulaire

Identifiant : INST-3

Concepts associés : instance de formulaire.

Préconditions : [INST-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Paul vient de [lister les instances de formulaires]. Il en sélectionne 2 et demande à les supprimer. Une confirmation lui est demandée. Il confirme la suppression. Les instances sont supprimées.

Scénario normal alternatif : Jean vient de [lister les instances de formulaires]. Il en sélectionne 2 et demande à les supprimer. Une confirmation lui est demandée. Il a changé d'avis et invalide sa demande de suppression. Il retourne à la [liste des instances de formulaires].

3.3.4 Éditer une Instance de Formulaire

Identifiant : INST-4

Concepts associés : instance de formulaire.

Préconditions : [INST-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Paul vient de [lister les instances de formulaires]. Il sélectionne un instance et demande à la modifier. Il retombe sur l'interface de [création d'une instance de formulaire]. La seule différence est que les champs titre, description, dates de début et de fin sont pré-remplis, prêts à être édités. Une fois les modifications faites, Paul demande à enregistrer la version retouchée.

Scénario normal alternatif : Paul vient de modifier une instance de formulaire, mais décide finalement de ne pas enregistrer la modification. Il lui suffit de changer de page. Un bouton lui propose d'annuler l'édition et de revenir à la [liste des instances de formulaires]. Il n'y a pas de sauvegarde automatique.

3.3.5 Remplir une Instance de Formulaire

Identifiant : INST-5

Concepts associés : instance de formulaire.

Préconditions : [MOD-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Zoé a reçu un lien menant à un formulaire web. Elle remplit les champs du formulaire et clique sur « Valider ». Un message lui indique que ses réponses ont bien été enregistrées.

Scénario d'erreur : Zoé a reçu un lien menant à un formulaire web. En arrivant sur la page, un message lui indique que le formulaire est désormais clos et qu'il fallait y répondre avant la date de fin de l'instance.

Scénario d'erreur alternatif : Zoé a reçu un lien menant à un formulaire web. En arrivant sur la page, un message lui indique que le formulaire n'est pas encore ouvert et lui donne la date à partir de laquelle elle pourra venir le compléter.

3.3.6 Visualiser les Résultats d'une Instance de Formulaire

Identifiant : INST-6

Concepts associés : instance de formulaire.

Préconditions : [INST-2]

Postconditions : -

Contraintes : -

Complexité : -

Scénario normal : Paul vient de [lister les instances de formulaires]. Il sélectionne un instance et demande à voir les résultats déjà rentrés. Une page lui affiche les résultats globaux et agrégés. Paul clique ensuite sur un lien lui permettant de voir l'ensemble des réponses individuelles. Ces réponses sont affichées.

Scénario d'erreur : Paul vient de [lister les instances de formulaires]. Il sélectionne un instance et demande à voir les résultats déjà rentrés. Un message lui indique que personne n'a encore répondu au formulaire.

Scénario d'erreur alternatif : Paul vient de [lister les instances de formulaires]. Il sélectionne un instance et demande à voir les résultats déjà rentrés. Un message lui indique que le formulaire n'a pas encore été ouvert au public (date de début dans le futur).

3.4 Tableau Associatif entre cas d'usage et tâches

Cas d'Usage VS. Tâches	
Cas d'Usage	Tâches
Modéliser un Formulaire	MOD-1, MOD-4
Gérer des Modèles de Formulaires	MOD-2, MOD-3
Instancier un Modèle de Formulaire	INST-1, INST-4
Gérer des Instances de Formulaires	INST-2, INST-3
Remplir une Instance de Formulaire	INST-5
Récupérer les Résultats d'une Instance de Formulaire	INST-6

Le découpage en tâches et leur description a permis de découper les cas d'usage et de préparer les actions utilisateurs nécessaires à leur réalisation. Il est maintenant possible de proposer des maquettes pour la partie graphique de l'application.

4 IHM Abstraite

4.1 Charte Graphique

Il n'y a pas de charte graphique à proprement parler.

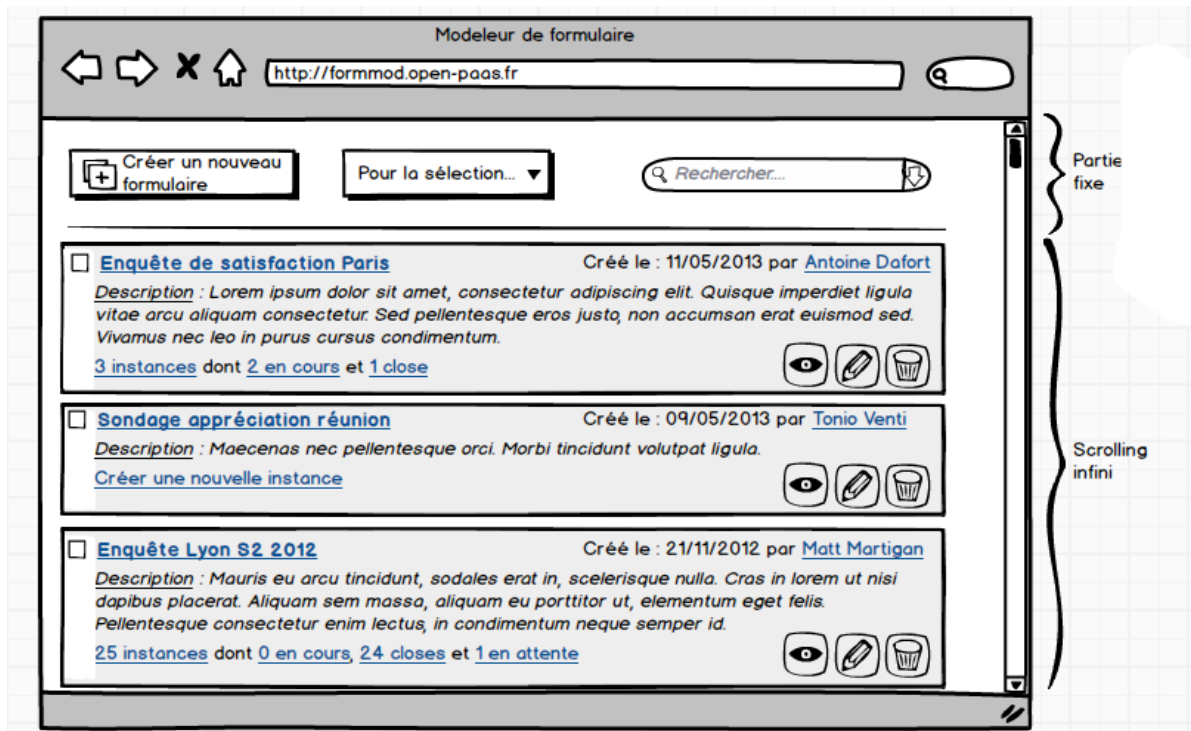
Cela dépend du contexte dans lequel va être utilisé cette application. Par exemple, dans le cadre d'Open Paas, la charte sera celle du portail de la plateforme.

La seule préconisation concerne la largeur de l'écran, qui doit être au minimum de 1024 pixels pour la création et la gestion de modèles et d'instances. Cette contrainte ne s'applique pas pour le remplissage d'une instance de formulaire.

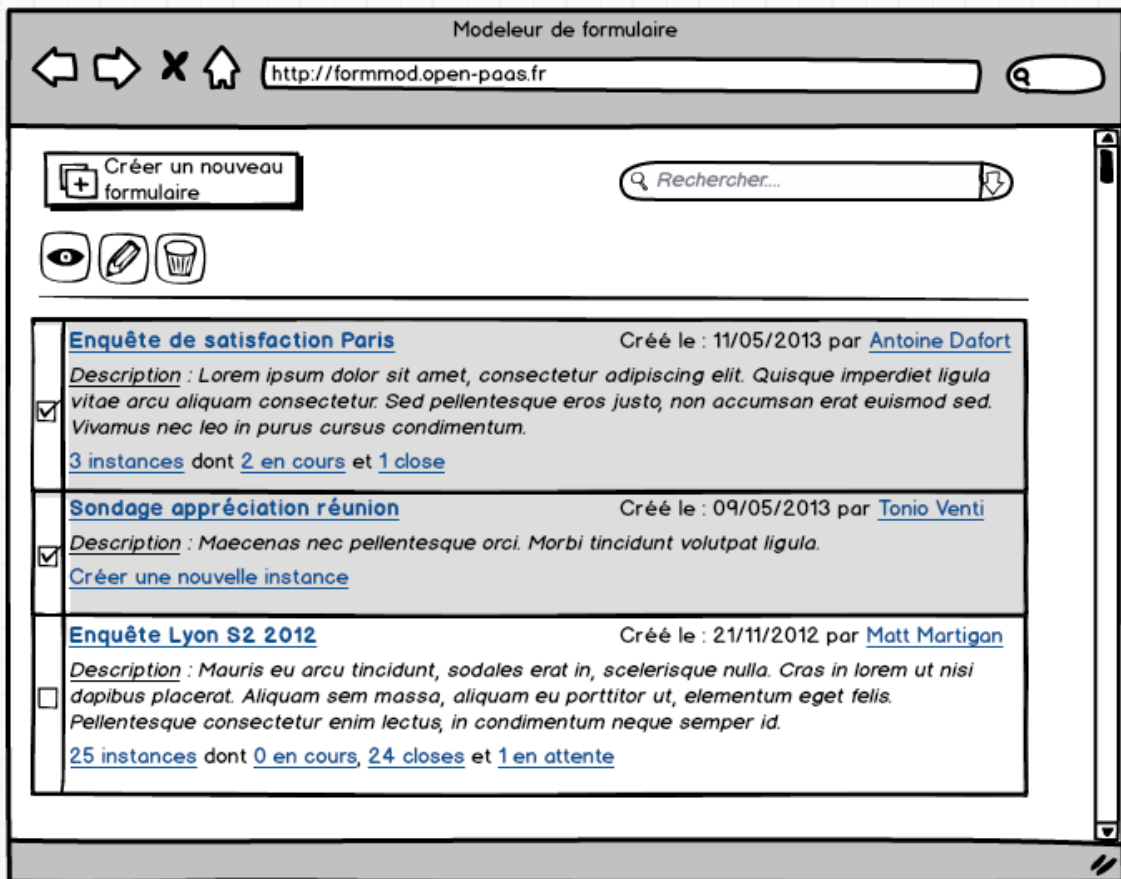
Plus globalement, la logique des écrans suit la logique des activités décrites en section 2.7. Les différentes pages sont listées ci-dessous.

- Une page qui liste les modèles de formulaires ou d'en créer un nouveau.
- Une page qui liste les instances de formulaires ou d'en créer une nouvelle.
- Une page qui permet de créer / éditer un modèle de formulaire.
- Une page qui permet de créer / éditer une instance de formulaire.
- Une page qui permet de remplir une instance de formulaire.
- Une page qui permet de visualiser les résultats d'une instance de formulaire.

4.2 Maquettes pour la liste des Modèles



[Accès à la liste des modèles](#)



Sélection de Modèles de Formulaires

Recherche avancée

Bouton flèche sur le moteur de recherche

Rechercher...

Titre :
 Auteur :
 Date de création : A partir du ▼ / /
 Description :

Recherche Avancée

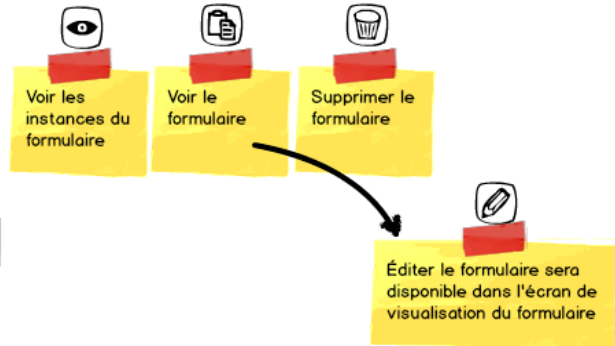
3 alternatives pour les actions sur les formulaires

I) Interface 3 boutons
Pas de possibilité de juste visualiser le formulaire, on tombe directement en mode édition

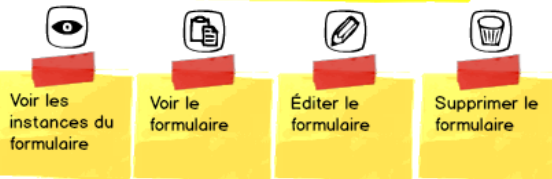


III) Interface 3 boutons
Et accès à l'édition en 2 étapes :

- 1 - Visualisation
- 2 - Édition



II) Interface 4 boutons



Actions disponibles pour les Modèles de Formulaires

Détails d'un bloc

Vers visualisation du formulaire

[Enquête Lyon S2 2012](#) Créé le : 21/11/2012 par [Matt Martigan](#)

Description : Mauris eu arcu tincidunt, sodales erat in, scelerisque nulla. Cras in lorem ut nisi dapibus placerat. Aliquam sem massa, aliquam eu porttitor ut, elementum eget felis. Pellentesque consectetur enim lectus, in condimentum neque semper id.

[25 instances](#) dont [0 en cours](#), [24 closes](#) et [1 en attente](#)

Vers fonctionnalités du RSE lié à l'identité (envoyer un message, voir le profil...) + filtre rapide : "lister tout les formulaires créés par cette personne"

Vers la liste des instances

Vers liste des instances avec filtre positionné sur la requête cliquée

Un Modèle de Formulaire

4.3 Maquettes pour les Instances de Formulaire

Firefox v

Nom : Enquête Paris
Description : une enquête de satisfaction pour nos clients parisiens.

Instance	Mise en ligne	Etat	Action
<input type="checkbox"/> Semestre 1 -	le 12/05/2013 par Toto	En Ligne	Accéder aux Résultats Editer Supprimer
<input type="checkbox"/> Semestre 2 -	le 09/05/2013 par Toto	Clôturé	Accéder aux Résultats Editer Supprimer

Actions Groupées ▼

Liste des Instances d'un Modèle

4.4 Maquettes pour la Modélisation d'un Formulaire

Firefox v

Sauvegarder **Fermer** **Mettre en Ligne**

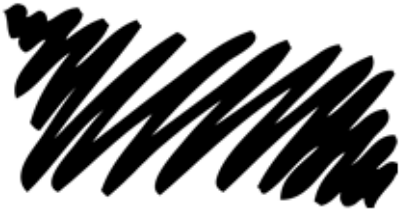
Nom :

Description :

Aperçu

Tab1

Tab2



La palette et la zone d'Assemblage d'un Modèle

5 Contraintes Techniques et Implémentation

5.1 Contraintes Techniques

- L'application doit produire des pages HTML valides (vérification W3C).
- Les feuilles de style CSS doivent être valides (vérification W3C, CSS 3 accepté, solutions propriétaires acceptées uniquement pour arrondir les bords).
- Interdiction d'utiliser des images pour faire des cadres ou des sections aux angles arrondis.
- L'application pourra utiliser du Javascript.

- L'application doit être internationalisée (français et anglais).
- L'application doit tourner sous Firefox, Chrome, IE 9, Safari et navigateurs sur tablettes. Le support de terminaux mobiles (smartphones) n'est à prévoir que pour l'affichage des instances à remplir (utilisateur = sondé).

- L'application peut être codée en Java, Scala ou Javascript (ce qui laisse un grand choix de frameworks possibles).
- L'application doit pouvoir être ajoutée à un serveur d'intégration continue (Jenkins).

- L'application doit stocker les résultats des formulaires, ainsi que la définition des modèles et des instances. Aucune contrainte n'est imposée au niveau de la spécification.

5.2 Gestion des Erreurs

Les scénarios mis en avant dans les tâches comportent des cas d'erreur.

Ces cas correspondent à des erreurs fonctionnelles. En revanche, ces scénarios n'intègrent pas la gestion des erreurs techniques (comme un perte de connexion à une base de données).

L'apparition d'une *stack* d'erreur (Java, Javascript) dans une page web est inacceptable. Les erreurs techniques doivent toutes être récupérées par l'application. Un message intelligible doit être adressé à l'utilisateur pour lui signifier qu'une erreur technique est survenue. On tentera, dans la mesure du possible, de fournir une explication simple sur la nature de cette erreur. Il sera proposé à l'utilisateur de rafraîchir la page d'ici quelques instants. Ce rafraîchissement peut même être effectué automatiquement par l'application ou le navigateur web.

Une trace plus avancée, pour le support, devra être laissée dans les logs de l'application.