



# Formulaire de contact

Yoann Beune & Elodie Linsac : groupe 2

# Fonctionnalités de l'application que nous avons développé

Fonctionnalité	MoSCoW
Interface graphique ergonomique proposant une gestion des contacts (CRUD sur les contacts)	M
Sauvegarde des informations de contact pour les recharger lorsque le logiciel se lance	M
Export d'un ou plusieurs contacts en <b>JSON</b>	M
Export d'un ou plusieurs contacts en <b>vCard</b>	M
Rechercher dans la liste des contact	S

and more ...

**L'application devra proposer une interface graphique ergonomique permettant de gérer une liste de contacts.**



Les opérations attendues sont les opérations CRUD :  
consultations d'une liste de contact(s), création de contact(s), modification  
de contact(s), suppression de contact(s)

Un contact est défini par les caractéristiques suivantes :

- nom (obligatoire)
- prénom (obligatoire)
- genre : homme/femme/non-binaire (obligatoire)
  - adresse (obligatoire)
- numéro de téléphone personnel (obligatoire)
  - une adresse email (obligatoire)
  - une adresse postale (obligatoire)
    - date de naissance (optionnel)
    - un pseudonyme (optionnel)
- numéro de téléphone professionnel (optionnel)
- un lien vers la page Github ou Gitlab du contact (optionnel)

## **Contraintes :**

**1) Veuillez à bien vérifier la validité des informations saisies par l'utilisateur pour les champs :**

- adresse email**
- lien vers la page Github ou Gitlab**

**2) Faut que ça biche**

**3) La sérialisation binaire, json et Vcard**

# **Analyse fonctionnelle :**

**1) Lecture du cahier des charges**

**2) User Persona**

**3) Backlog**

# User persona :

Michelle : Cheffe de projet de 47 ans d'une équipe de développeurs junior, ayant pour passion la gastronomie italienne, le crochet et le sport. Cherchant à obtenir un répertoire efficace et ergonomique des informations de chacun de ses collaborateurs avec lesquels elle travaille afin d'éviter de perdre du temps à leurs demander ces informations là sur les heures de sprint lorsqu'elle s'occupe de faire les groupes de projet.

Elle cherche un moyen rapide et pratique qui saura lui donner les informations dont elle a besoin quand elle en a besoin pour former le plus efficacement possible des groupes de travail qui peuvent être performants et terminer les sprint au plus vite.

Le but de l'application est de répertorier chaque personnes qui travaillent pour Michelle à travers plusieurs attributs comme leurs noms, prénoms, adresse etc ..., dans laquelle elle aura le choix de soit l'enregistrer en Json, soit en Vcard qui ressemblera à une carte de visite, de pouvoir modifier le contact choisi au cas où et/ou de pouvoir le supprimer

Bien sûr toutes les données seront enregistrées sur le disque dur

De plus il y aura un menu déroulant pour le genre, une vérification du format de l'email et le format du lien de sa page github , l'option de rechercher quelqu'un dans la liste donnée.

Aussi l'interface graphique des informations d'une personne sera sous la forme d'une carte de visite dans le cas où elle choisit l'enregistrement de données en Vcard.

# Backlog :

A faire :

1. Réfléchir aux différents aspects de la fenêtre et à ces fonctionnalités tout en restant ergonomique
2. Faire des semblants de maquette histoire d'avoir un idée du format de la fenêtre
3. Faire le persona afin d'avoir une ligne à suivre
4. Faire vérifier par le PO le persona et les pseudos maquettes
5. Faire une interface graphique ergonomique proposant une gestion des contacts  
(CRUD sur les contacts)
1. Créer une sauvegarde des informations de contact pour les recharger lorsque le logiciel se lance
2. Donner la possibilité de tout exporter ou d'exporter qu'une sélection de contacts choisis
3. Créer un export d'un ou plusieurs contacts en JSON
4. Créer un export d'un ou plusieurs contacts en vCard
5. Penser à la possibilité de rechercher une personne dans la liste des contact
6. Merge les deux codes afin que les boutons soit associés à la fenêtre principale
7. Corriger les erreurs de dépendances, d'import et de concordance de code
8. Corriger la forme de l'enregistrement en Json et en Vcard
9. Mettre en place la sérialisation en binaire
10. Mettre en place la barre de recherche afin de rendre la recherche de contact plus facile si nous avons le temps
11. Si on a le temps, améliorer l'aspect design de la fenêtre utilisateur



# **Maquettage de l'application**

**1) Zoning**

**2) Wireframe**

**3) Mockup**

# Zoning :

Barre de recherche

Tableau contact

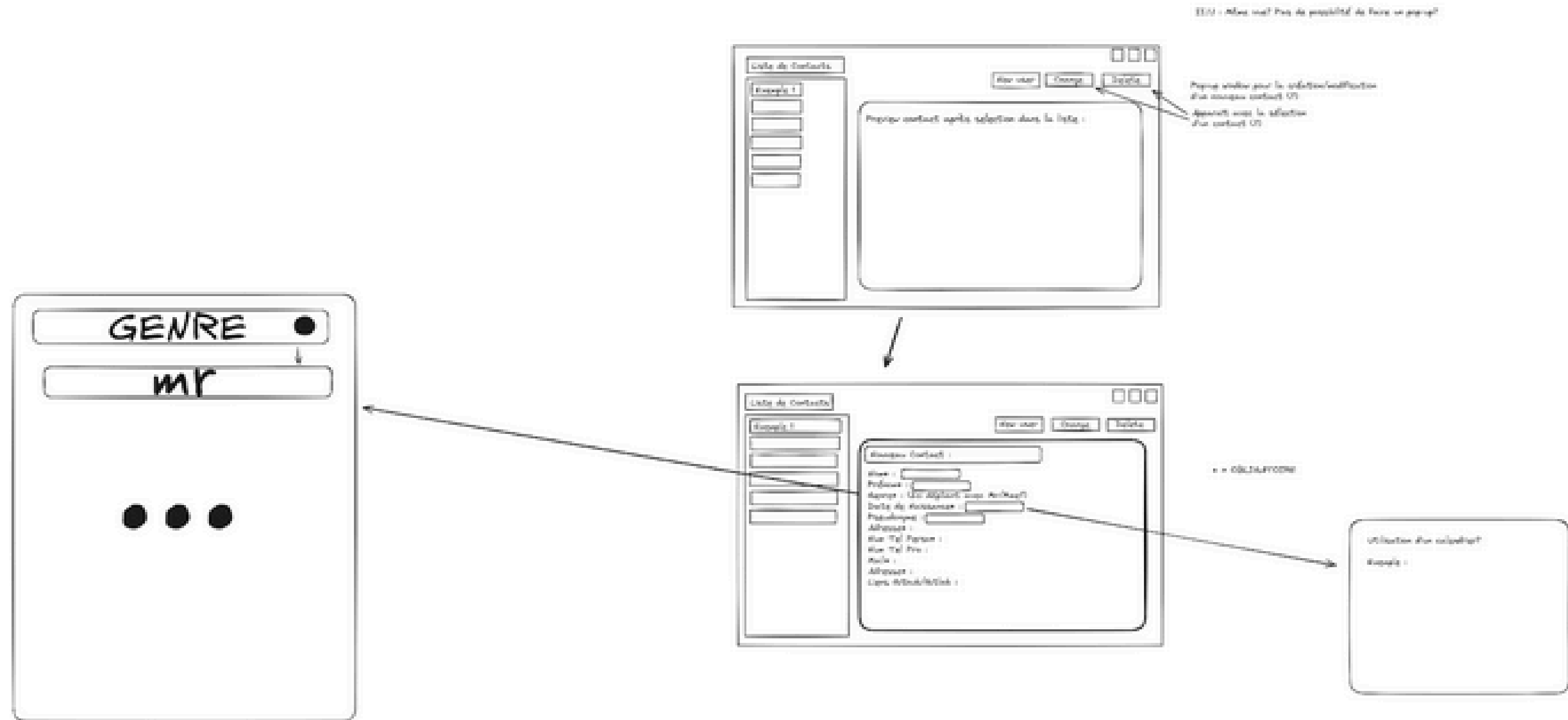
Créer

Changer

Supprimer

Formulaire Contact

# Wireframe



# Mockup/maquette haute fidélité

The mockup shows a contact management application window titled "MainWindow.fxml". It features a search bar at the top left, a table with columns "Last Name" and "First Name" (currently empty with the message "Aucun contenu dans la table"), and a form on the right for adding or editing contact details. The form includes fields for Last Name, First Name, Surname, Gender, Date of Birth, Address, Zip Code, Personal Phone, Professional Phone, Mail, and GitHub / GitLab. Action buttons like "Export Selection", "Export All", "New Contact", "Delete", and "Change" are also present.

MainWindow.fxml

searchField

Last Name	First Name
Aucun contenu dans la table	

Export Selection

Export All

New Contact

Last Name :

lastNameField

First Name :

firstNameField

Surname :

surnameField

Gender :

Date of Birth :

adressField

Address :

zipCodeField

Zip Code :

personalPhoneField

Personal Phone :

professionalPhoneField

Professional Phone :

mailField

Mail :

gitField

GitHub / GitLab :

Delete

Change

# Exemple de Vcard d'un des nos tests :

```
≡ contacts2024-08-07.vcard
1  BEGIN:VCARD
2  VERSION:4.0N:hghhrh;hrhrh;Select your gender.
3  FN:hrhrh hghhrh
4  ADR:gegeeg;12345
5  EMAIL:elkg@gmail.com
6  TEL:3214;
7  ORG:
8  END:VCARD
9
10
```

# Exemple de Json d'un de nos tests :

```
"firstName": "Bob",  
"lastName": "Bob",  
"surName": "Bob",  
"gender": "Man",  
"birthDate": "2000-05-05",  
"address": "Bob",  
"zipcode": "11111",  
"personalPhone": "0606060606",  
"professionalPhone": "",  
"mail": "Bob@bob.bob",  
"gitLink": ""
```

# UML donné par le cahier des charges

