

Fiche d'investigation de fonctionnalité

| | |
|--|--------------------------|
| Fonctionnalité : Filtrer les recettes | Fonctionnalité #2 |
| Problématique : Afin d'assurer une expérience optimale, nous visons à développer une fonctionnalité de recherche pour trouver une recette qui répond aux besoins de l'utilisateur parmi les recettes disponibles. | |

| | |
|---|--|
| Option 1 : Recherche avec Boucles natives Dans cette option, nous utilisons les boucles natives de Javascript. L'avantage est qu'elle est simple à implémenter et flexible, mais l'inconvénient principal est qu'elle peut être moins optimisée pour de grandes collections de recettes, affectant potentiellement les performances de recherche et l'expérience utilisateur. | |
| Avantages <ul style="list-style-type: none">⊕ Simple à implémenter⊕ Flexible | Inconvénients <ul style="list-style-type: none">⊖ Non optimisé pour les grandes collections de recettes⊖ Diminution des performances de recherche et de l'UX |
| Nombre de champs minimum à remplir pour la recherche : 1 | |
| Nombre de champs minimum à sélectionner pour filtrer : 1 | |

| | |
|--|---|
| Option 2 : Recherche avec Programmation Fonctionnelle Dans cette option, on utilise la programmation fonctionnelle avec les méthodes de l'objet Array de JavaScript (foreach, filter, map, reduce). Elle offre une meilleure performance et une lisibilité du code, mais peut être plus complexe à comprendre pour les développeurs non familiers avec la programmation fonctionnelle. | |
| Avantages <ul style="list-style-type: none">⊕ Les méthodes de l'objet Array sont généralement plus rapides que les boucles natives.⊕ Code plus lisible et plus facile à maintenir.⊕ Meilleure expérience utilisateur | Inconvénients <ul style="list-style-type: none">⊖ Complexité de la mise en œuvre pour les débutants⊖ Avantage non significative pour les petites collections de recettes par rapport à la première option |
| Nombre de champs minimum à remplir pour la recherche : 1 | |
| Nombre de champs minimum à sélectionner pour filtrer : 1 | |

| |
|--|
| Solution retenue : Nous avons retenu la deuxième version utilisant la programmation fonctionnelle. La raison est que cette approche offre une recherche plus rapide et un code plus lisible, améliorant ainsi l'expérience utilisateur. L'utilisateur bénéficie d'une recherche précise et efficace, rendant la navigation sur le site plus naturelle et agréable. |
|--|

Annexes

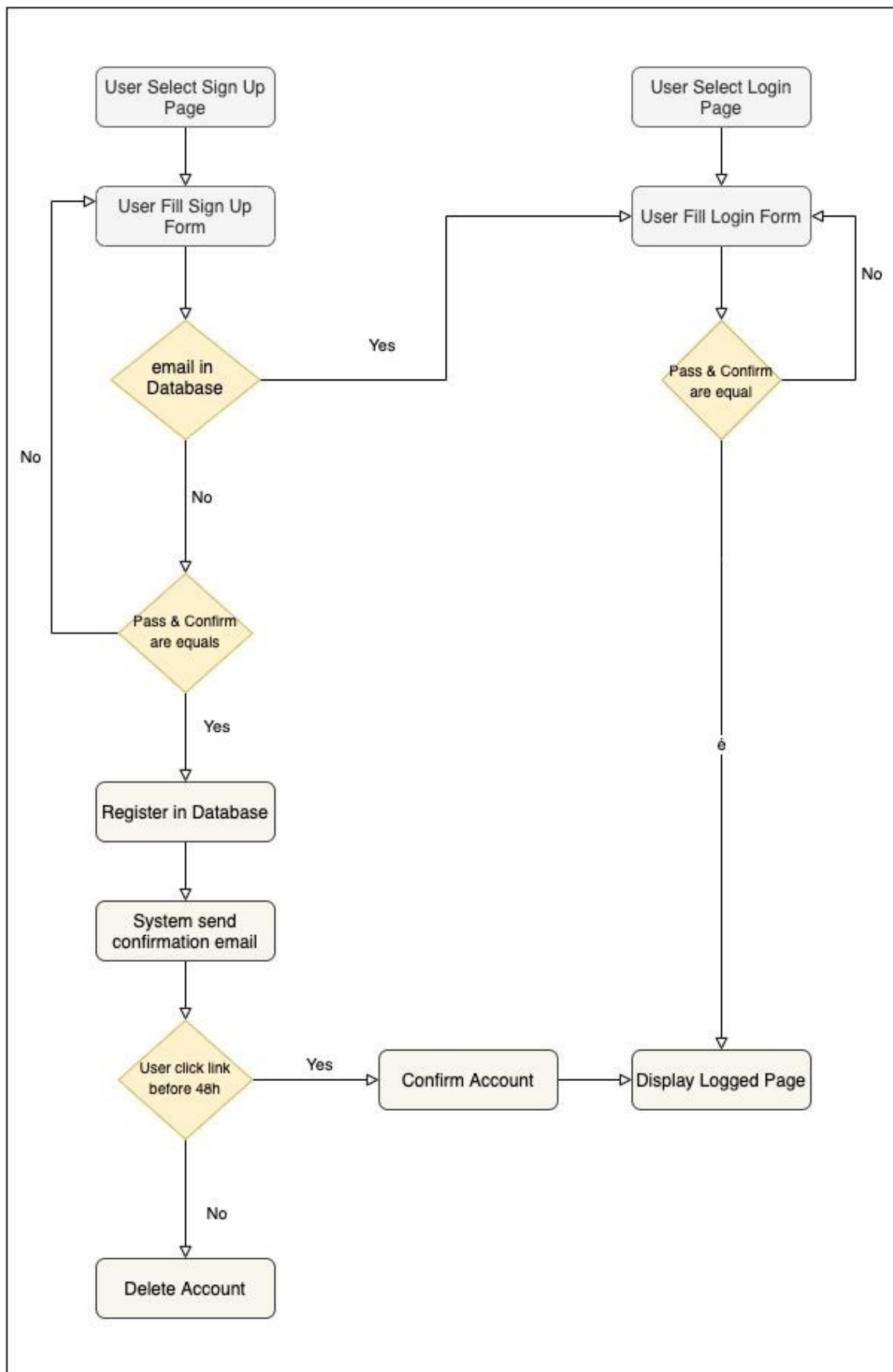


Figure 1 - Diagramme d'activité Classic login Workflow

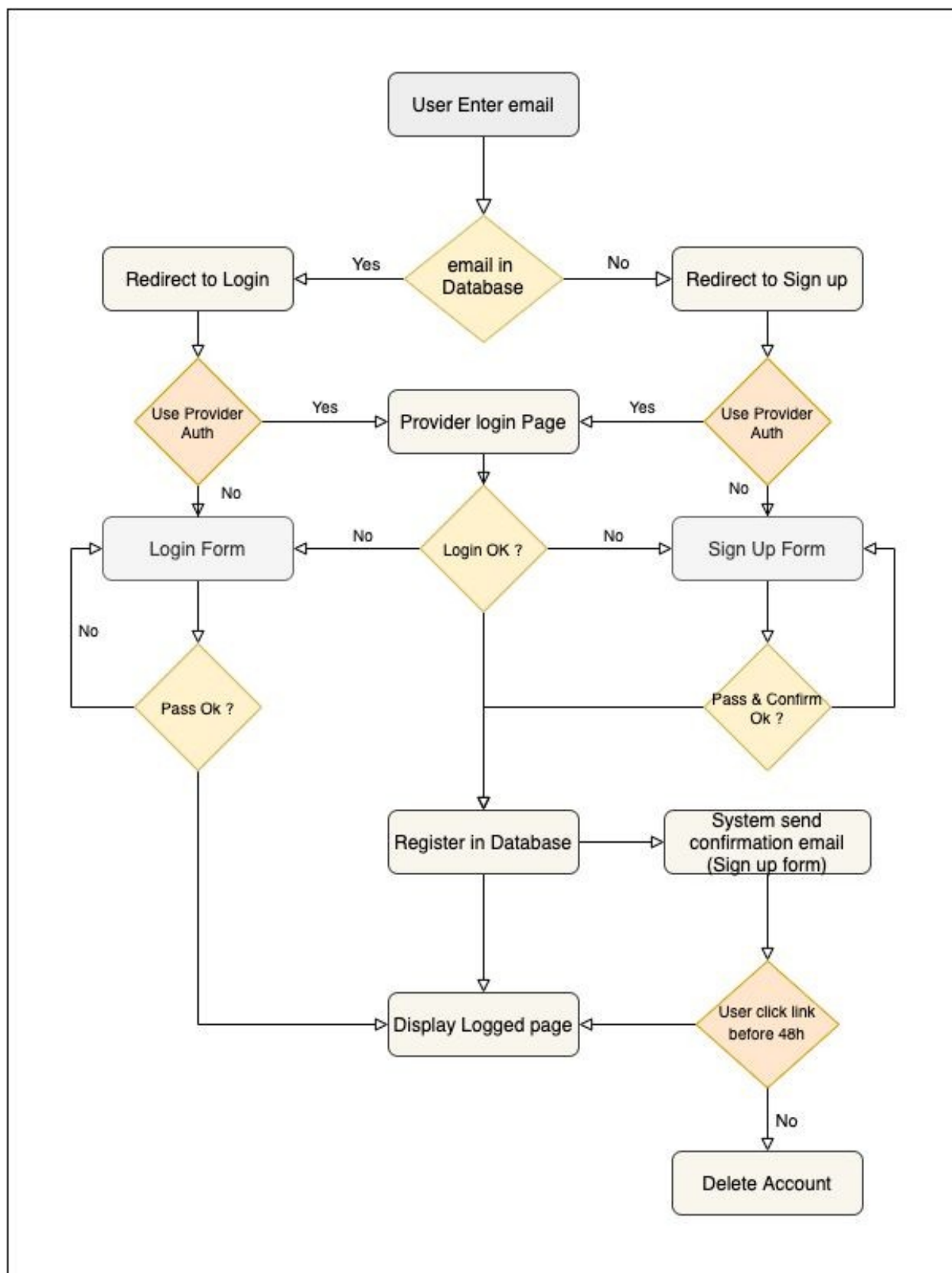


Figure 2 : Approche "Email First" et usage de Google Identity Toolkit