

# Guide d'étapes clés

Designez une application Python adaptée aux besoins clients

## Comment utiliser ce document ?

Ce guide vous propose un découpage du projet en étapes. Vous pouvez suivre ces étapes selon vos besoins. Dans chacune, vous trouverez :

- des recommandations pour compléter la mission ;
- les points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet (attention, celui-ci peut varier d'un apprenant à l'autre).

Suivre ce guide vous permettra ainsi :

- d'organiser votre temps ;
- de gagner en autonomie ;
- d'utiliser les cours et ressources de façon efficace ;
- de mobiliser une méthodologie professionnelle que vous pourrez réutiliser.

**Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et sera différente selon votre vitesse de progression.**

## Recommandations générales

Nous avons décomposé ces étapes clés afin que vous puissiez terminer une page à la fois, étape par étape :

1. La page de connexion.
2. La page Interface de chat.
3. La page du calendrier.
4. La page de gestion des tâches.
5. La page Tableau de bord.

À l'aide de ces étapes, vous complétez d'abord les user stories pour une page, mettez à jour le tableau Kanban en conséquence, puis mettez à jour ou créez les wireframes en fonction de ces user stories. Ce n'est pas

la seule méthode pour réaliser ce projet, mais c'est très utile. Lors d'une tâche réelle, vous prendrez probablement au moins les wireframes une page à la fois.

Pour le diagramme de cas d'usage UML, nous vous conseillons de créer une version initiale au début du projet, en ayant lu le projet et les notes de la réunion lancement du projet. Ce diagramme peut fournir un aperçu de tout le système pendant que vous travaillez sur chaque page individuelle. Ensuite, vous pouvez mettre à jour le diagramme à la fin du projet pour tenir compte des changements apportés.

Les étapes clés se décomposent en un ensemble idéalisé. Cependant, il est normal de se rendre compte qu'au fur et à mesure que vous progressez dans les tâches, vous pouvez avoir besoin de réviser les étapes précédentes, surtout si vous comprenez mieux le projet. Mettez cette approche en pratique dans ce projet.

## Points de vigilance

- **Incohérences entre les différents documents :** Lorsque l'on rédige une grande quantité de documents, on oublie fréquemment de prendre en compte la modification d'un livrable dans les autres. Ce type d'erreur peut être interprété comme un réel manque de professionnalisme lors d'une présentation au client.
  - Par exemple, lors de la création de vos wireframes alors que vous travaillez sur un écran particulier, vous pouvez modifier un élément commun à d'autres écrans, comme un élément de navigation.
  - Veuillez ne pas oublier de mettre à jour le même élément sur les autres pages afin d'éviter de créer des incohérences qui pourraient troubler le client quant à la conception que vous lui présentez pour approbation.
- **Orthographe :** Faites attention aux fautes d'orthographe dans le texte et les wireframes. Celles-ci peuvent ternir votre image professionnelle.
- **Nommages des éléments :** Veillez à utiliser un nommage clair et cohérent afin de réduire les suppositions (et les malentendus ou la confusion) sur ce que représente un élément particulier.
  - Par exemple, un mot de passe doit être appelé "mot\_de\_passe" plutôt que "mdp" ou une autre abréviation ambiguë.

## Ressources

Gardez en tête que la plupart des ressources sur la méthodologie agile sont rédigées en anglais.

Ces ressources sont utiles pour toutes les étapes liées à la rédaction **de user stories et de critères d'acceptation** :

- Le cours OpenClassrooms [Initiez-vous à la gestion de projet agile](#), en particulier le chapitre suivant :
  - Partie 3, Chapitre 1 : [Inspectez les User Stories](#)
- Les chapitres suivantes du cours OpenClassrooms [Gérez votre projet avec une équipe Scrum](#) (plus avancé) :
  - Partie 3, Chapitre 1 : [Confirmez la maturité d'une User Story](#)
  - Partie 3, Chapitre 2 : [Explicitiez la conformité d'une User Story](#)
- Les ressources externes rédigées en anglais :
  - [Agile alliance explain on User Stories](#)
  - [Agile alliance explain on 'Given - When - Then'](#)
  - [Refine User Stories and Acceptance Criteria with Agile](#)
  - [Given, When, Then — Agile Acceptance Criteria](#)

Ces ressources sont utiles pour toutes les étapes liées à la création de **wireframes** :

- Le cours OpenClassrooms [Concevez une interface cliquable](#), en particulier les chapitres suivants :
  - Partie 1, Chapitre 2 : [Découvrez les prérequis pour produire des maquettes](#)
  - Partie 1, Chapitre 3 : [Réalisez des wireframes](#)
  - Partie 2, Chapitre 1 : [Réalisez un prototype](#)
  - Partie 2, Chapitre 2 : [Testez un prototype](#)
- Les ressources externes rédigées en anglais :
  - [What Exactly Is Wireframing? A Comprehensive Guide](#)
  - [A Beginner's Guide to Wireframing in UX Design](#)
  - [Prototyping 101: The Difference between Low-Fidelity and High-Fidelity Prototypes and When to Use Each](#)

## Étape 1 : *Mettre en place le Kanban et créer un diagramme initial de cas d'usage UML.*

15% de progression

---

### **Avant de démarrer cette étape, je dois avoir :**

- Lu le projet entier, incluant la correspondance avec le client et les notes de la réunion lancement du projet.

### **Une fois cette étape terminée, je devrais avoir :**

- Mis à jour le tableau Kanban (sur Notion, Trello ou GitHub).
- Créé un diagramme initial de cas d'usage UML pour l'application.

### **Recommandations :**

- Vous pouvez utiliser le [kanban](#) déjà fourni sur Notion ou mettre en place un tableau kanban sur Trello ou GitHub.
  - Quel que soit le logiciel que vous utilisez pour le kanban, vous ne devrez remplir qu'une seule colonne, ANALYZE.
- Le diagramme UML de cas d'usage est un aperçu de la manière dont les acteurs utiliseront l'application et interagiront les uns avec les autres. Restez simple et visuellement clair !
- Dans le diagramme UML de cas d'usage :
  - Les acteurs sont représentés par des bâtons (veillez à inclure tous les utilisateurs qui interagissent avec le système).
  - Les cas d'usage sont représentés par des cercles ou des ellipses.
  - Assurez-vous de représenter les relations entre les cas d'usage.
  - Faites également attention aux relations "extend" et "include" !

### **Points de vigilance :**

- Vous n'avez besoin de faire qu'un seul diagramme UML de cas d'usage pour toute l'application.
  - Pour les détails de chaque page, vous utiliserez les user stories, les critères d'acceptation et les wireframes.

## Ressources :

- Le cours OpenClassrooms [Appliquez le principe du Domain-Driven Design à votre application](#), en particulier:
  - Partie 1, Chapitre 2: [Résolvez des problèmes de conception avec les modèles de domaines](#).
  - Partie 2, Chapitre 1: [Formalisez votre modèle de domaine avec un diagramme de cas d'utilisation](#).
  - Partie 2, Chapitre 3: [Appliquez le principe du Domain-Driven Design à votre application](#).
- [Diagramme de cas d'utilisation](#) (rédigé en anglais)

Étape #2 : *Compléter les User Stories et les critères d'acceptation de la page de connexion.*

25% progression

---

## Avant de démarrer cette étape, je dois avoir :

- Revu les notes de la réunion lancement du projet pour la page de connexion.
- Créé un diagramme initial de cas d'usage UML pour l'application.
- Mis à jour le tableau Kanban.

## Une fois cette étape terminée, je devrais avoir :

- Une liste des User Stories et critères d'acceptation pour la page de connexion.

## Recommandations :

- Analysez les informations contenues dans les notes de projet et décomposez-les en différentes actions que l'utilisateur doit pouvoir effectuer sur la page de connexion.
- Les notes de projet contiennent un aperçu de haut niveau de la fonctionnalité pour la page de connexion, votre tâche consiste donc à traduire ces exigences dans le format correct des user stories.
- Les exigences du client ne contiennent pas souvent tous les détails dont vous avez besoin pour rédiger des user stories complètes. Utilisez les notes du projet et vos propres expériences avec diverses applications et sites Web pour rédiger les user stories.

- Vous devez également demander des précisions au client si vous estimez que des détails supplémentaires sont nécessaires ou si les exigences données sont vagues.
- Pour ce projet, votre mentor peut se substituer au client : vous pouvez lui poser les questions que vous poseriez à un client pour clarifier ses besoins.

### Points de vigilance :

- Les user stories doivent se concentrer sur la description de la fonctionnalité (ce que l'utilisateur doit être capable de faire).
  - Les user stories ne doivent **pas** contenir de détails de mise en œuvre (c'est-à-dire comment la fonctionnalité sera implémentée).
- N'oubliez pas que chaque user story doit inclure tous les critères d'acceptation qui, selon vous, permettront de valider que l'user story a été correctement implémenté lors de tester la fonctionnalité.
- Idéalement, les critères d'acceptation devraient au moins prendre en compte les scénarios d'utilisation normaux, c'est-à-dire que l'utilisateur se comporte comme prévu et sans aucune erreur.
  - Cependant, les utilisateurs font parfois des erreurs et ne se comportent pas toujours comme prévu.
    - Par exemple, ils peuvent saisir un mauvais mot de passe ou entrer des lettres dans un champ qui devrait contenir des chiffres.

Lorsque vous rédigez les critères d'acceptation, essayez de penser à ces scénarios et à la façon dont des critères d'acceptation bien rédigés peuvent aider à détecter les erreurs et à rendre la fonctionnalité plus robuste.
- N'oubliez pas d'utiliser le format correct pour les user stories et les critères d'acceptation, comme indiqué dans les ressources ci-dessus.
  - Pour les user stories : En tant que <utilisateur> → je veux <fonctionnalité> → afin de (pour) <raison> → fonction valide si <critère>.
  - Pour les critères d'acceptation : Format Étant donné - Quand - Alors (Given - When - Then).
- Vous pouvez ajouter les user stories à la colonne ANALYZE du tableau Kanban maintenant ou à l'étape suivante.

## Étape #3 : Compléter le wireframe et le tableau kanban pour la page de connexion.

35% progression

---

### **Avant de démarrer cette étape, je dois avoir :**

- Un jeu complet des User Stories et critères d'acceptation pour la page de connexion.

### **Une fois cette étape terminée, je devrais avoir :**

- Mis à jour le wireframe de la page de connexion.
- Mis à jour le tableau kanban par les user stories et les tâches pertinentes de la page.

### **Recommandations :**

- Le wireframe initiale (fournie dans les notes de réunion du projet) contient des éléments liés à certaines des fonctionnalités requises pour cette page, mais pas toutes.
  - Pour chaque user story, examinez la conception du wireframe pour voir s'il comporte les éléments d'interface utilisateur (UI) nécessaires pour permettre à l'utilisateur d'effectuer la tâche du story. Si ce n'est pas le cas, ajoutez-les.
- Travailler sur les wireframes vous permettra de "parcourir" et de valider les user stories, vous devez donc aussi considérer cela comme une chance d'itérer sur les stories que vous avez rédigés et de les améliorer.
  - Y a-t-il une fonctionnalité décrite dans les user stories qui n'a pas été implémentée dans le wireframe ?
  - Y a-t-il quelque chose dans le wireframe qui n'est pas décrit dans les user stories ? Si c'est le cas, y a-t-il un user story manquant, ou y a-t-il un élément d'UI pas nécessaire dans le wireframe ?
- Si vous n'avez pas encore ajouté les user stories à la colonne ANALYZE du tableau Kanban, faites-le maintenant.

## Étape #4 : Compléter les User Stories et les critères d'acceptation de la page de chat.

45% progression

---

### **Avant de démarrer cette étape, je dois avoir :**

- Revu les notes de la réunion lancement du projet pour la page d'interface de chat

### **Une fois cette étape terminée, je devrais avoir :**

- Une liste des User Stories et critères d'acceptation pour la page d'interface de chat.

### **Recommandations :**

- Analysez les informations contenues dans les notes de la réunion du projet et décomposez-les en différentes actions que l'utilisateur doit pouvoir effectuer et en informations qu'il doit voir sur la page de l'interface de chat.
- Les notes de la réunion lancement du projet contiennent un aperçu de haut niveau des informations et des fonctionnalités souhaitées que vous devrez inclure dans la page d'interface de chat. Votre tâche consiste donc à traduire ces exigences dans le format correct de l'user story.

## Étape #5 : Compléter le wireframe et le tableau kanban pour la page de chat.

50% progression

---

### **Avant de démarrer cette étape, je dois avoir :**

- Un jeu complet des User Stories et critères d'acceptation pour la page de chat.

### **Une fois cette étape terminée, je devrais avoir :**

- Mis à jour le wireframe de la page de chat.
- Mis à jour le tableau kanban par les user stories et les tâches pertinentes de la page.



**Recommandations :**

- Le wireframe initiale (fournie dans les notes de réunion du projet) contient des éléments liés à certaines des fonctionnalités requises pour cette page, mais pas toutes.
- Pour chaque user story, examinez la conception du wireframe pour voir s'il comporte les éléments d'UI nécessaires pour permettre à l'utilisateur d'effectuer la tâche du story. Si ce n'est pas le cas, ajoutez-les.

## Étape #6 : *Compléter les livrables pour la page du calendrier.*

**65% progression**

---

**Avant de démarrer cette étape, je dois avoir :**

- Revu les notes de la réunion lancement du projet pour la page du calendrier.

**Une fois cette étape terminée, je devrais avoir :**

- Une liste des User Stories et critères d'acceptation pour la page du calendrier.
- Un wireframe de la page du calendrier.
- Mis à jour le tableau kanban par les user stories et les tâches pertinentes de la page.

**Recommandations :**

- À ce stade, n'hésitez pas à combiner les étapes des user stories, ses critères d'acceptation, du Kanban et du wireframe en une seule étape clé. Vous pouvez itérer entre le travail sur les user stories et la mise à jour du wireframe et du tableau Kanban.
- Les notes de la réunion lancement du projet pour cette page sont très brèves. Cependant, les clients fournissent régulièrement des exigences de très haut niveau pour des fonctionnalités considérées comme bien connues et pas particulièrement uniques à l'application.
  - Par exemple, vous savez, d'après les notes, qu'une "page de calendrier classique" est nécessaire, ce qui implique que vous savez quelles informations et quelles fonctionnalités une "page de calendrier classique" contient.

- Cet ensemble de fonctionnalités est le premier qui n'a pas d'un wireframe de départ, c'est donc l'occasion d'en développer un à partir de zéro. Toutefois, veillez à réutiliser tous les éléments des wireframes précédents qui sont communs à celui-ci.

#### **Points de vigilance :**

- Ce scénario peut vous obliger à valider soigneusement toutes les hypothèses que le client - ou vous-même - a sur ces fonctionnalités d'une plateforme.
  - Par exemple, quelles sont les façons d'afficher les informations du calendrier ?
  - Lesquelles sont classiques et lesquelles ne le sont pas ? Y a-t-il des informations sur les événements qui sont propres à cette plateforme ?
- Une fois que vous aurez commencé à explorer les détails, vous constaterez qu'une exigence considérée comme classique comprendra, en fait, des détails spécifiques aux besoins du client.

### *Étape #7 : Compléter les livrables pour la page de gestion de tâches.*

#### **75% progression**



#### **Avant de démarrer cette étape, je dois avoir :**

- Revu les notes de la réunion lancement du projet pour la page de gestion de tâches.

#### **Une fois cette étape terminée, je devrais avoir :**

- Une liste des User Stories et critères d'acceptation pour la page de gestion de tâches.
- Un wireframe de la page de gestion de tâches.
- Mis à jour le tableau kanban par les user stories et les tâches pertinentes de la page.

#### **Recommandations :**

- Les exigences pour la page de gestion des tâches introduisent l'idée de rôles d'utilisateur : différents utilisateurs ont accès à différentes fonctionnalités en fonction de leur type, ou rôle, d'utilisateur.
  - Pour cette page, vous devez vous assurer que les user stories

précisent le rôle de l'utilisateur auquel l'user story s'applique (étudiants ou bénévoles).

- Comment clarifier la différence entre les stories pour l'utilisateur "général" (c'est-à-dire les user stories qui décrivent la fonctionnalité disponible pour tous les utilisateurs, quel que soit leur rôle), et ceux pour les utilisateurs ayant un rôle spécifique ? Comment cela affecte-t-il la façon dont vous rédigez vos critères d'acceptation ?
- Posez-vous les questions suivantes :
  - Étant donné que cette partie de la plateforme comporte des fonctionnalités qui diffèrent en fonction du type d'utilisateur (étudiants ou bénévoles), comment le wireframe va-t-il refléter cela ?
  - En quoi l'interface utilisateur est-elle différente pour chaque type d'utilisateur ?

## Étape #8 : *Compléter les livrables pour la page de tableau de bord*

**90% progression**

---

### **Avant de démarrer cette étape, je dois avoir :**

- Revu les notes de la réunion lancement du projet pour la page de tableau de bord.
- Une liste des User Stories et critères d'acceptation pour les autres pages.
- Wireframes mis à jour pour les autres pages.

### **Une fois cette étape terminée, je devrais avoir :**

- Une liste des User Stories et critères d'acceptation pour la page de tableau de bord.
- Un wireframe de la page de tableau de bord.
- Mis à jour le tableau kanban par les user stories et les tâches pertinentes de la page.

### **Recommandations :**

- Le tableau de bord résume et rassemble les informations provenant d'autres parties de la plateforme.
  - Lorsque vous rédigez les user stories pour cette page, pensez à la quantité d'informations appropriée pour un tableau de

bord : elle doit être suffisante pour fournir l'aperçu dont l'utilisateur a besoin, mais pas trop pour ne pas être écrasante.

- De même, vous devrez examiner les détails des user stories des autres pages pour trouver les informations que vous pourriez vouloir inclure dans les user stories du tableau de bord.

## Étape #9 : *Vérifier tout et préparer les livrables.*

### 100% progression

---

#### **Avant de démarrer cette étape, je dois avoir :**

- Un jeu complet d'user stories et de critères d'acceptation pour toutes les pages.
- Un jeu complet de wireframes pour toutes les pages.
- Un tableau Kanban complet avec toutes les user stories et leurs tâches connexes.

#### **Une fois cette étape terminée, je devrais avoir :**

- Vérifié et exporté des user stories, des critères d'acceptation et des wireframes dans un format portable (PDF).
- Mis à jour du diagramme UML des cas d'usage afin de décrire toutes les interactions possibles des utilisateurs avec la plateforme.
- Exporté du diagramme UML dans un format portable (PDF).

#### **Recommandations :**

- Maintenant que vous avez rédigé toutes les user stories et tous les critères d'acceptation pour les pages du site, vous pouvez mettre à jour votre diagramme UML de cas d'usage pour les refléter.
  - Veillez à ajouter les acteurs, les cas d'usages ou les relations entre les cas d'usages (telles que "extend" ou "include") qui manquaient dans la version initiale du diagramme en fonction de votre travail dans les étapes intermédiaires.
  - N'oubliez pas de représenter les acteurs par des bâtons et les cas d'utilisation par des cercles ou des ellipses.
- Vérifiez la cohérence de vos livrables :
  - Avez-vous été cohérent dans le nommage des champs et des éléments dans toutes les user stories sur toutes les pages ?

- La disposition des éléments ou des sections de l'interface utilisateur (par exemple, la navigation) communs à plus d'une page est-elle la même sur toutes ces pages ?
- Avez-vous utilisé le même format recommandé pour toutes les user stories et les critères d'acceptation ?

**Projet terminé !**