

# Budy

Problème : Difficulté de bien s'occuper de ses plantes sans avoir une visualisation de leur environnement de vie, d'irrégularités quant à cet environnement de vie, ou de personnes vers lesquelles se tourner pour demander des conseils.

Solution : Via une application téléphone, faire un suivi de l'environnement de vie de ses plantes, avoir un aperçu de l'évolution de cet environnement de vie au cours du temps, et pouvant mettre en place des alertes déclenchées par une anomalie dans cet environnement pour une meilleure surveillance. Avoir la possibilité d'en discuter directement avec ses proches/amis/connaissances de manière centralisée sur l'application, avec les données à disposition si souhaité (pour donner/demander des conseils, ou simplement avoir un sujet de discussion commun).

Définitions :

- Budy : Nom de l'application, vient d'un jeu de mot entre « Bud », ou « Bourgeon », et « Buddy », ou « Copain », pour signifier la partie sociale de l'application.
- Budies : Il s'agit là du nom qu'on utilise pour les capteurs qui servent à surveiller les plantes. Ils devraient être plantés dans la terre de la plante, selon le manuel fournis avec le capteur lors de l'achat.
- Alerte : Règle qui définit quand on envoie des incidents.
- Incident : Information quant au déclenchement d'une alerte.

## Exigences fonctionnelles

1. Un utilisateur devrait avoir la capacité de se créer un compte sur l'application à l'aide d'un nom d'utilisateur unique, d'une adresse email unique, et d'un mot de passe.
2. Un utilisateur devrait pouvoir se connecter à l'application via son adresse email et son mot de passe.
3. Un utilisateur devrait être capable de changer son mot de passe, son nom d'utilisateur, et de se rajouter une/retirer son/changer son image de profile.
4. Un utilisateur devrait pouvoir se déconnecter.
5. Un utilisateur devrait avoir un aperçu rapide de ses plantes dans une liste sur une page dédiée, où des informations basiques sur chaque plante et leur état devrait être mentionné.
6. Un utilisateur devrait pouvoir générer une entrée dans sa liste de plantes
  - 6.1. Obligatoire : lui attribuer un nom unique (unique parmi sa propre liste, plusieurs utilisateurs peuvent avoir le même nom de plante).
  - 6.2. Optionnel : renseigner le type de la plante parmi celles de notre database, si présente, afin d'obtenir des alertes préconfigurées
7. Un utilisateur devrait pouvoir filtrer/trier la liste de plantes selon plusieurs critères (« A un budy assigné », « Ordre alphabétique », etc.).
8. Un utilisateur devrait pouvoir, depuis cette liste, accéder à une page présentant les détails quant à la plante, ainsi que des options. Ces détails et options devraient inclure :
  - 8.1. Image(s), si assignée(s), ou/et de quoi ajouter une image
  - 8.2. Son nom
  - 8.3. Ses statistiques actuelles (selon informations récoltées)
  - 8.4. Des graphes de ses statistiques au cours du temps
  - 8.5. De quoi ajouter un budy si aucun budy n'est assigné, ou de quoi libérer le budy assigné s'il y en a un
  - 8.6. De quoi supprimer la plante de sa liste
    - 8.6.1. Si un budy est associé, le libère
    - 8.6.2. Les données des statistiques sont supprimées
    - 8.6.3. L'image est supprimée
  - 8.7. Configurer les alertes (voir plus loin pour les détails des alertes)
9. Un utilisateur devrait être capable, en ayant un budy pas encore assigné, de l'enregistrer à une plante (chaque budy ne peut être lié qu'à une plante à la fois).
10. Pour chaque plante ayant un budy assigné, l'application devrait faire de la collecte de données de manière régulière (configurable, pas inférieur à fréquence min.), et de stocker ces données pour une utilisation et un affichage ultérieur.
11. L'utilisateur devrait pouvoir, pour chaque plante, créer des alertes.
  - 11.1. Une alerte peut être active ou inactive. Une alerte ne peut être active que s'il y a un budy assigné. Une alerte peut être rendue inactive par l'utilisateur, supprimée par l'utilisateur, ou rendue active si elle respecte la condition ci-dessus.
  - 11.2. Quand une alerte active se déclenche, elle envoie une notification d'incident à l'appareil sur lequel l'application est installée (si l'application a le droit d'envoyer des notifications). Elle rajoute aussi dans tous les cas la notification au stockage interne à l'application.
  - 11.3. L'alerte est définie par un nom unique (par plante), et des conditions d'activation vérifiées périodiquement. Elle a aussi une sévérité.
12. Un incident contiendra toutes les informations quant au déclenchement d'une alerte. Ceci inclut :
  - 12.1. La plante concernée
  - 12.2. Le nom de l'alerte
  - 12.3. La sévérité de l'alerte
  - 12.4. La période d'activation (début, et si applicable, fin)

13. Un utilisateur devrait pouvoir aller sur la partie communautaire de l'application, et voir les postes de ses amis, sous la forme d'une liste contenant des informations minimales sur chacun des postes.
14. Un utilisateur devrait avoir l'option de faire des demandes d'amitié, et d'accepter ou refuser de telles demandes qui lui seraient faites.
15. Un utilisateur devrait pouvoir supprimer des gens de sa liste d'amis.
16. Recevoir une demande, la réception d'une réponse à une demande faite, ou se faire retirer de la liste d'amis de quelqu'un devrait envoyer une notification sur lequel l'application est installée (si l'application a le droit d'envoyer des notifications). Elle rajoute aussi dans tous les cas la notification au stockage interne à l'application.
17. Un utilisateur devrait pouvoir mettre un poste comme favori.
18. Un utilisateur devrait pouvoir filtrer pour ne voir que ses postes mis en favoris (ajout d'autres options de filtrage plus tard).
19. Un utilisateur devrait pouvoir
20. Un utilisateur devrait pouvoir aller sur la page d'un de ces postes, et obtenir le poste détaillé, contenant toutes les informations du poste. Ceci pourrait inclure :
  - 20.1. Un titre au poste
  - 20.2. Une image reliée
  - 20.3. Une plante reliée
  - 20.4. Un message
  - 20.5. Les commentaires et réponses au poste
21. L'utilisateur devrait pouvoir commenter sur un poste, ou répondre à un commentaire, sous la forme de texte.
22. L'utilisateur devrait pouvoir supprimer son commentaire ou sa réponse, laissant un message à la place semblable à « Message supprimé ».
23. L'utilisateur recevra des notifications sur l'appareil sur lequel l'application est installée (si l'application a le droit d'envoyer des notifications) si/quand un autre utilisateur répond à un de ses commentaires/à une de ses réponses. Elle rajoute aussi dans tous les cas la notification au stockage interne à l'application.
24. L'utilisateur devrait pouvoir ajouter un poste, contenant toutes ou seulement certaines des informations mentionnées ci-dessus (obligatoire : titre, message ; optionnel : plante reliée, image).
25. L'utilisateur devrait recevoir des notifications sur l'appareil sur lequel l'application est installée (si l'application a le droit d'envoyer des notifications) si/quand un autre utilisateur commente sur le poste. Elle rajoute aussi dans tous les cas la notification au stockage interne à l'application.
26. L'utilisateur devrait pouvoir supprimer son poste, supprimant toutes les informations, commentaires, réponses, images, etc. y étant relié.
27. L'utilisateur devrait avoir une visualisation sur les notifications reçues dans l'application, permettant un accès rapide aux informations concernées.

## Exigences non-fonctionnelles

1. Le groupe de travail doit être de 3 ou 4 personnes (4)
2. Le groupe doit documenter sa composition dans un fichier excel disponible dans le canal « Général » Teams de l'équipe PDG Classroom 2025 (Groupe 9)
3. Le travail doit être effectué dans un délai de 3 semaines
4. Le projet doit inclure les étapes intermédiaires et rendus aux dates indiquées dans la présentation initiale
  - 4.1. Semaine 1
    - 4.1.1. Description du projet (objectif, requirements fonctionnels, requirements non fonctionnels)
    - 4.1.2. Description préliminaire de l'architecture
    - 4.1.3. Mockups (Figma, papier-crayon, etc.)
    - 4.1.4. Landing page mise en ligne
    - 4.1.5. Description des choix techniques
    - 4.1.6. Description du processus de travail (p.ex. : git flow, devops, etc.)
    - 4.1.7. Mise en place des outils de développement (Issue tracker ... etc.)
    - 4.1.8. Mise en place d'un environnement de déploiement
    - 4.1.9. Mise en place d'une pipeline de livraison et de déploiement (CI/CD)
    - 4.1.10. Démonstration du déploiement d'une modification
  - 4.2. Semaine 3
    - 4.2.1. Description du problème et de la solution
    - 4.2.2. Code source du projet
    - 4.2.3. Instructions reproductibles pour lancer le projet en local
    - 4.2.4. Instructions reproductibles pour déployer une nouvelle fonctionnalité avec la pipeline de CI/CD
    - 4.2.5. Instructions de contribution au projet (quel est le workflow adopté, comment quelqu'un d'externe peut contribuer)
    - 4.2.6. Vidéo de présentation de 1-4 minutes
    - 4.2.7. Présentation et démonstration en classe la dernière semaine (15 minutes, dont 5 minutes de questions et la vidéo)
5. Le groupe doit être à disposition pour travailler à 100% durant ces 3 semaines
6. Le groupe doit rester en contact avec le groupe des enseignants et assistants pour un suivi de leur travail
  - 6.1. Les groupes doivent, chaque matin, remplir le formulaire de point journalier à rendre dans le canal Teams créé pour chaque groupe
7. Les outils utilisés devraient minimiser les coûts du projet, sinon ne pas engendrer de coûts du tout. Certaines ressources sont à disposition des groupes.
8. La gestion de l'authentification des utilisateurs devrait être gérée selon les pratiques classiques de la cybersécurité.
9. Autres mesures de cybersécurité générales à déterminer pour empêcher de potentielles attaques sur l'application.