

Plan de tests

Application de compagnon virtuel sur smartphone



Étudiants : Mathis Ruffieux et Néréïs Dugaleix

Encadrante : Mme. Lydie du Bousquet

Tuteur : Mr. Damien Pellier

Informations d'identification du document

Référence du document	D7
Version du document	1.01
Date du document	13/06/2022
Auteurs	Mathis Ruffieux et Néréïs Dugaleix

Éléments de vérification du document

Validé par	Mme. Lydie du Bousquet
Validé le	
Soumis le	13/06/2022
Type de diffusion	Document électronique (pdf)
Confidentialité	Non confidentiel

Sommaire

I) Introduction

II) Ensemble de tests ou recettes ainsi que leur ordonnancement

III) Description des tests d'intégration

Pour chaque test :

- Description du test : Décrire le test de façon externe
- But du test : Décrire ce que prouve le test
- Principe de réalisation : Décrire la procédure de test ainsi que les paramètres et

IV) Description des tests unitaires

Pour chaque module/classe :

- Mise en œuvre
- Pour chaque jeu d'essai : Données d'entrées, résultats attendues, critère d'évaluation

I) Introduction

II) Ensemble de tests

Les tests d'intégrations se basent sur des fonctionnalités directes de l'application, en suivant l'ordonnancement des fonctions appelées afin de les tester une à une.

Ces tests sont découpés en quatre catégories :

- Les tests sur le chat, visant à tester les fonctionnalités des scénarios et des boutons du chat.
- Les tests sur le journal, vérifiant les CRUD des différentes pages et le passages d'une page secrète à non secrète.
- Les tests sur les options pour modifier chacune d'entre elles et leur impact global sur l'application.
- Les tests sur le premier démarrage et redémarrage de l'application.

III) Tests d'intégration

1) Envoie d'une réponse fermée

Description du test :

L'envoi d'une réponse fermée s'effectue après une question fermée (Annexe 1).

But du test :

Le test permet de voir si le message d'une réponse fermée s'écrit bien dans les bases SQL et si après son envoi le robot répond bien avec la réplique suivante attendue selon un scénario donné (Annexe 3).

Principe de réalisation :

On clique sur une réponse fermée (Annexe 1).

Ce bouton déclenche la fonction "displayClosedReply" en lui passant un objet Reply en paramètre.

Cette fonction ajoute le message dans la BD avec la fonction "addMessage" et appelle ensuite l'ID de la réplique suivante du robot inscrit dans l'objet Reply avec la fonction "displayQuestion".

Cette nouvelle fonction postera les messages du robot dans la BD et affiche la réponse suivante attendue par l'utilisateur (réponse fermée, ouverte ou fin du scénario).

Après ces opérations la page du chat est rafraîchie permettant ainsi de récupérer en base les nouveaux messages et de les afficher à l'écran.

Le test est un succès lorsque ces nouveaux messages sont affichés correctement, car ils ont été récupérés depuis la BD (Annexe 2).

2) Envoie d'une réponse ouverte

Description du test :

L'envoi d'une réponse ouverte s'effectue après une réponse ouverte

But du test :

Le test permet de voir si le message d'une réponse ouverte s'écrit bien dans les bases SQL et si après son envoi le robot répond bien avec la réplique suivante attendue selon un scénario donné (Annexe 4).

Principe de réalisation :

On clique sur la réponse ouverte (Annexe 4).

Ce bouton déclenche la fonction "displayOpenReply" avec cette fois non pas un objet Reply mais seulement le String écrit par l'utilisateur.

De même, cette fonction écrit en BD le message et récupère l'ID de la question suivante du robot dans l'objet Question précédent, pour appeler la fonction "displayQuestion".

Comme avec une question fermée, la page est ensuite rafraîchie permettant ainsi de récupérer en base les nouveaux messages et de les afficher à l'écran.

Le test est un succès lorsque ces nouveaux messages sont affichés correctement, car ils ont été récupérés depuis la BD (Annexe 5).

3) Aimer un message ou le passer en secret

Description du test :

On like ou on passe un message en secret.

But du test :

L'objet message concerné doit changer son booléen isLiked ou isSecret en BD.

Ce changement est visible depuis l'interface de l'application avec un icône de cœur ou de clef (Annexe 6).

Principe de réalisation :

On clique sur l'icône de cœur ou de clef. Cet icône doit changer de couleur ce qui montre qu'il a bien été sélectionné (dans l'annexe 6, le cœur est sélectionné).

Ensuite, on clique sur un message.

On peut s'assurer que l'objet message sélectionné a bien été modifié car dans le widget ChatBox, pour chaque message, on vérifie si le booléen isLiked et isSecret sont à TRUE pour afficher ou non un icône de cœur et un icône de clef. Le test est donc un succès si ces icônes apparaissent sur les bon messages (Annexe 6).

4) Suppression d'un message

Description du test :

On supprime un message.

But du test :

Le message doit être supprimé dans la BD. Pour vérifier cela la page est rafraîchie et le message ne doit plus apparaître.

Principe de réalisation :

On clique sur l'icône poubelle du chat, puis on clique sur le message à supprimer. La fonction "delete" est alors appelée avec l'ID du message, et le message est supprimé en BD.

Ensuite la page du chat est rafraîchie, appelant à nouveau la liste de messages. Si le message n'est pas affiché, alors il n'existe plus.

5) Création d'une page au journal

Description du test :

Ajouter une nouvelle page au journal avec un titre et une description.

But du test :

Vérifier que l'ajout s'effectue correctement et que la note est conservée après réouverture de l'application.

Principe de réalisation :

On clique sur le "+" en bas à droite de l'écran journal pour ouvrir la création d'une page (Annexe 7).

Si on retire le titre (ajouté automatiquement avec l'heure de la création de la page), un message d'erreur doit apparaître indiquant que le champ est vide (Annexe 8).

Lorsqu'on sauvegarde la nouvelle page, elle s'ajoute bien dans les nouvelles pages du journal (Annexe 9).

Si on redémarre l'application, cette nouvelle page est toujours présente ce qui confirme qu'elle a bien été écrite en base.

6) Passage d'une page en secret / non-secret

Description du test :

Passer une note en secret.

Retirer une note des secrets.

But du test :

Vérifier que les notes non secrètes et secrètes sont affichées dans leurs pages respectives et qu'il est possible de les faire passer d'une page à l'autre en les cochants / décochants.

Principe de réalisation :

Cocher une note non secrète pour la faire passer en secret et inversement.

Les pages secrètes ne sont affichées que dans la page dédiée aux secrets (Annexe 10).

7) Suppression et modification des pages

Description du test :

On modifie une page du journal et on supprime une autre page.

But du test :

On vérifie que la modification et la suppression fonctionnent.

Principe de réalisation :

On clique sur une note pour la modifier ce qui ouvre une page dédiée (Annexe 11).

On enregistre la modification et on vérifie sur le journal qu'elle a bien été effectuée.

De même, on supprime une note et on vérifie qu'elle n'est bien plus présente.

Pour chacune de ces opérations on vérifie que la snackbar (petite barre grisée en bas) s'affiche et se retire au bout d'un temps défini, indiquant ainsi à l'utilisateur que l'opération a été effectuée avec succès (Annexe 10).

(La fonction "showSnackBar" est écrite dans le fichier utils.dart et appelle le package "material.dart".)

8) Modification des informations personnelles

Description du test :

Dans la page des paramètres, on modifie les informations suivantes :

Le nom, le prénom, la date de naissance et l'heure des notifications.

But du test :

Vérifier que les informations ont été bien modifiées en base et puissent être récupérées.

Principe de réalisation :

Sur la page des options, on modifie le nom, le prénom, la date de naissance et l'heure des notifications puis on clique sur valider.

Sur le bouton valider sont appelées les fonctions “setVariable” permettant de modifier chaque variable en BD. Ces variables sont ensuite récupérées lorsqu’on quitte et ouvre à nouveau la page des options avec la fonction “getVariableByName”. Le test fonctionne pour le nom et le prénom mais pas pour la date de naissance ni l’heure des notifications, car le format de date n’est pas géré en String (Nous n’avons pas encore implémenté de fonction convertissant la date en String pour l’enregistrer en base).

Ces variables, tel que le nom et le prénom sont ensuite utilisées par le robot lorsqu’il nous appelle par notre nom.

Par exemple ici (Annexe 14), le robot nous appelle par notre prénom modifié (“Jean 2”).

9) Modification de la langue

Description du test :

Modifier la langue depuis les options et la tester

But du test :

Tester si l’application est prise en charge dans les 3 langues : Anglais, Français et Japonais.

Principe de réalisation :

Depuis les options, cliquer sur un autre drapeau puis cliquer sur valider pour changer la langue.

Tester ensuite l’application de chat et du journal dans la nouvelle langue (Annexe 15) pour voir si tous les boutons ont été traduits.

A l’heure actuelle les scénarios ne sont pas encore écrits en japonais. Il suffit de compléter le fichier Json avec les 3 langues pour que cela fonctionne.

10) Exporter les données

Description du test :

Exporter l’historique du chat et du journal à des fins d’analyse.

Attention, ce test ne fonctionne pour l’instant que sous Android.

But du test :

Pouvoir récupérer un fichier Json sur le téléphone d’un utilisateur, sans passer par internet.

Principe de réalisation :

Depuis la page des options, cliquer sur le bouton Exporter.

Si c’est la première fois, l’application demandera à l’utilisateur les droits en écriture.

Deux fichiers Json sont alors générés sur le téléphone, l’un pour les messages et l’autre pour le journal.

Les messages catégorisés comme secrets, ainsi que les pages du journal secrètes sont volontairement omises.

Brancher le téléphone et récupérer les deux fichiers Json nouvellement créés depuis le répertoire Download.

Vérifier que les objets des messages et des pages du journal sont présents.

11) Importer les données

Description du test :

Importer de nouveaux scénarios via un fichier Json.

Attention, ce test ne fonctionne pour l'instant que sous Android.

But du test :

Tester si les mises à jour des scénarios fonctionnent et s'effectuent correctement.

Principe de réalisation :

Créer un fichier Json en suivant la documentation pour écrire ce fichier correctement dans le manuel d'utilisation.

Placer ce fichier dans le répertoire Download du téléphone.

Ouvrir l'application, le menu des options, puis cliquer sur le bouton Importer.

Si c'est la première fois, l'application demandera à l'utilisateur les droits en lecture.

Retourner dans le chat et lancer un scénario pour voir s'il fonctionne bien, et s' il a été mis à jour (en fonction des modifications apportées au fichier Json).

Pour vérifier que les scénarios fonctionnent bien on peut réitérer les tests du chat (1) et (2).

12) Test du premier démarrage

Description et but du test :

Tester si le scénario de bienvenue est appelé et s'effectue correctement.

Principe de réalisation :

Si l'application a déjà été lancée ultérieurement, aller dans les réglages du téléphone et supprimer les données (Annexe 16) ce qui permettra de faire comme si c'était le premier démarrage.

Lancer l'application et vérifier que le scénario de bienvenue est bien appelé.

De plus, l'application est par défaut en anglais. La première question du robot est donc de choisir la langue (Annexe 17).

Continuer le scénario de bienvenue jusqu'à la fin.

Ensuite, aller dans les options et vérifier que le nouveau nom et nouveau prénom entré sont bien affichés.

S'ils sont affichés correctement, les variables ont été enregistrées en base correctement et le test est un succès.

13) Test de redémarrage d'un scénario

Description et but du test :

Tester si lorsqu'on ferme l'application au milieu d'un scénario, ce dernier reprend à l'endroit où on l'est arrêté.

Principe de réalisation :

Démarrer un scénario, puis quitter l'application.

Ouvrir l'application et vérifier qu'on en est à la même question du scénario.

L'indice de la question posée est enregistré en base au fur et à mesure de l'avancement d'un scénario.

Avec la fonction "initLifeCycle", cet indice est lu à l'ouverture de l'application. Si il est non vide c'est qu'un scénario était en cours, on reprend donc là où on s'est arrêté avec la fonction "resumesOngoingScenario".

Ce test permet donc de tester toutes ces fonctionnalités.

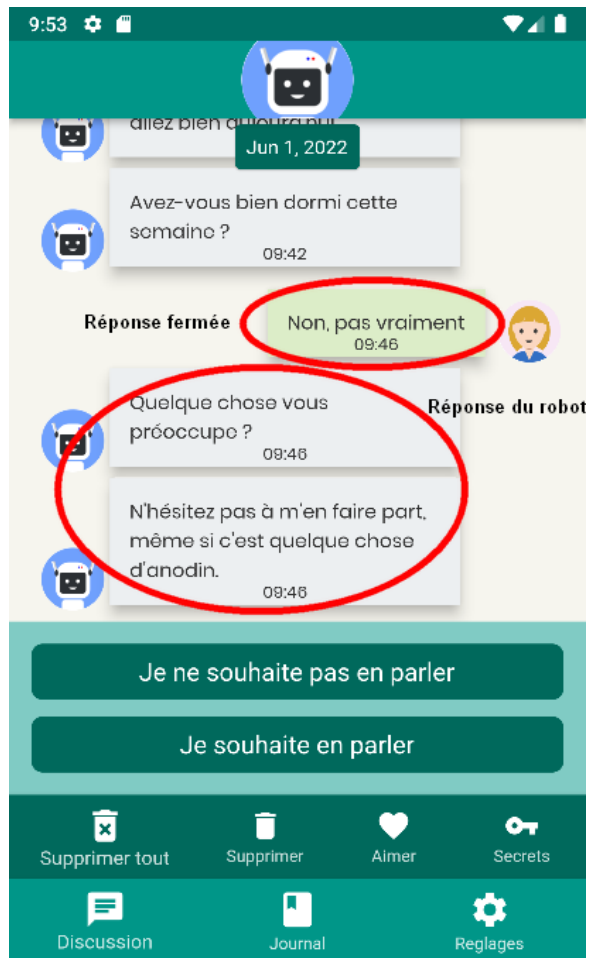
IV) Tests unitaires

-> à compléter

V) Annexes

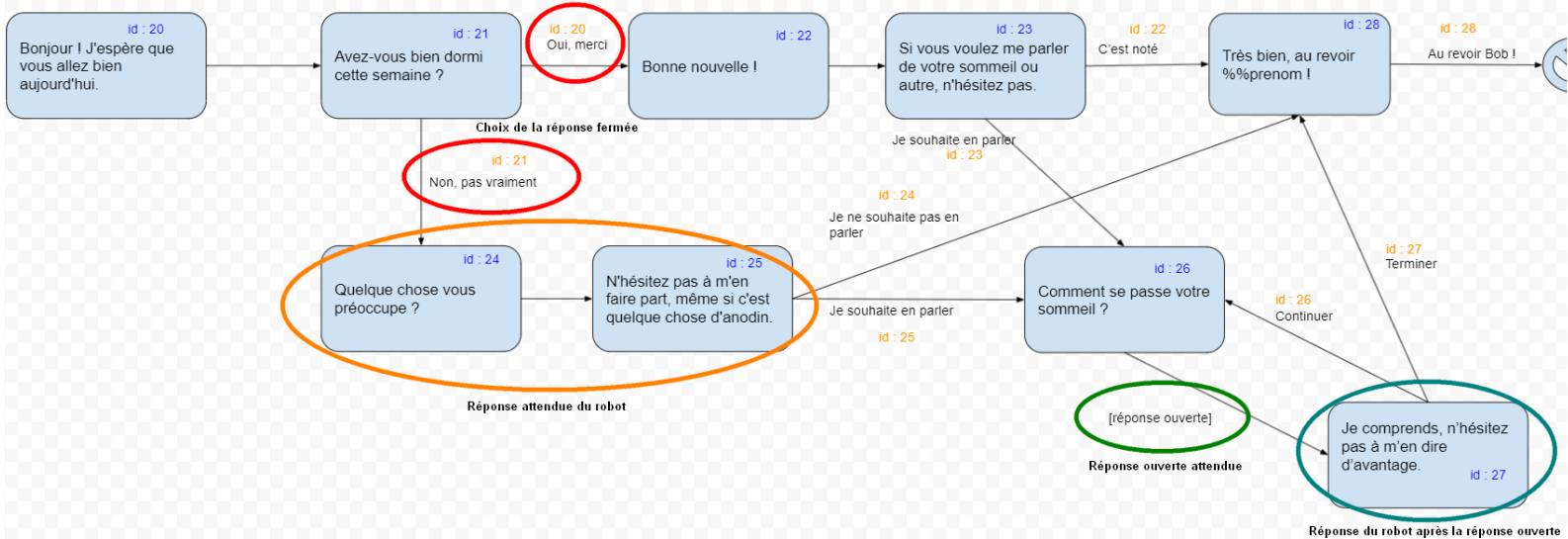
Annexe 1 : Réponse fermée attendue

Annexe 2 : Réponse fermée postée

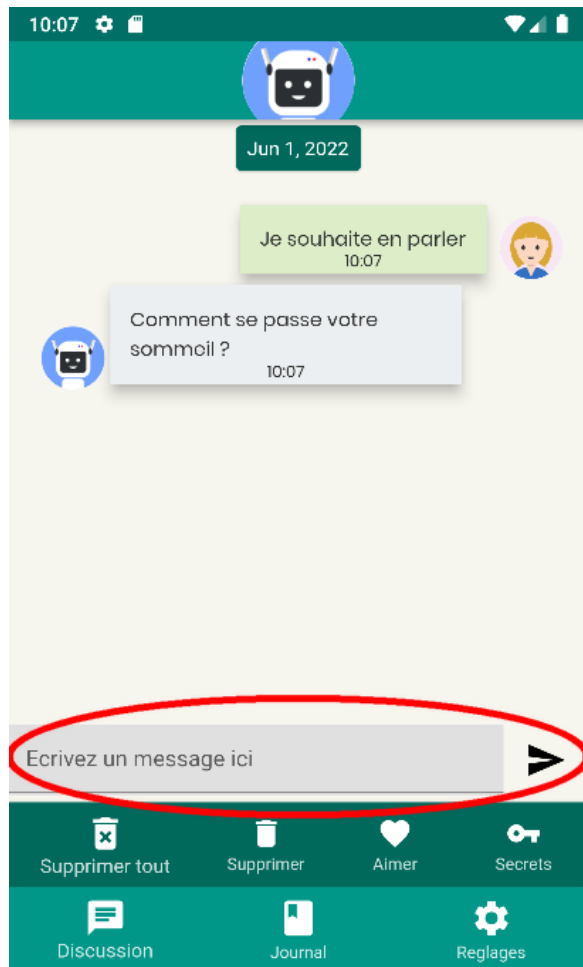


Annexe 3 : Réponses attendues du scénario testé

Scénario 2



Annexe 4 : Réponse ouverte attendue



Annexe 5 : Réponse de l'utilisateur

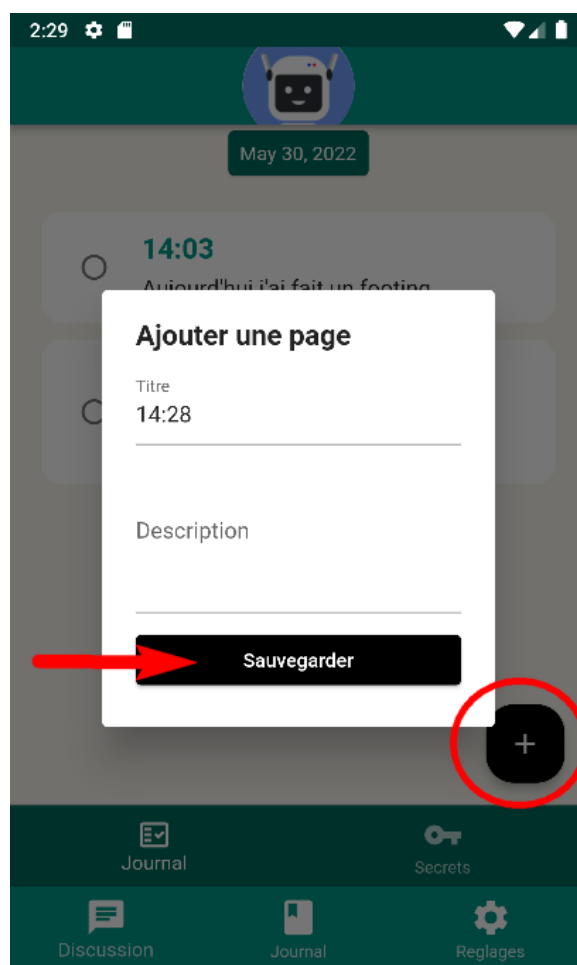


Annexe 6 : Aimer ou message secret

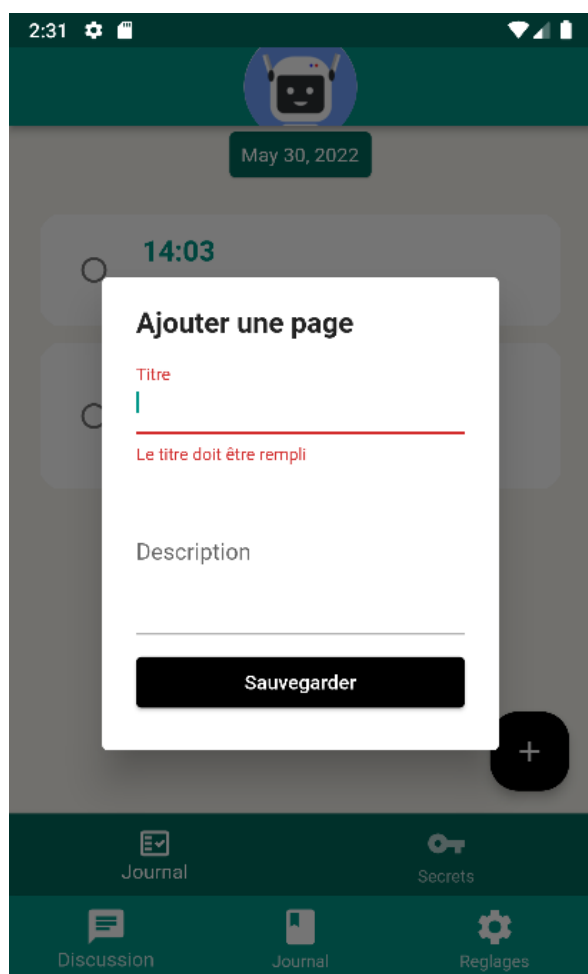
Annexe 7 : Ajout d'une page au journal



Annexe 8 : Message d'erreur journal



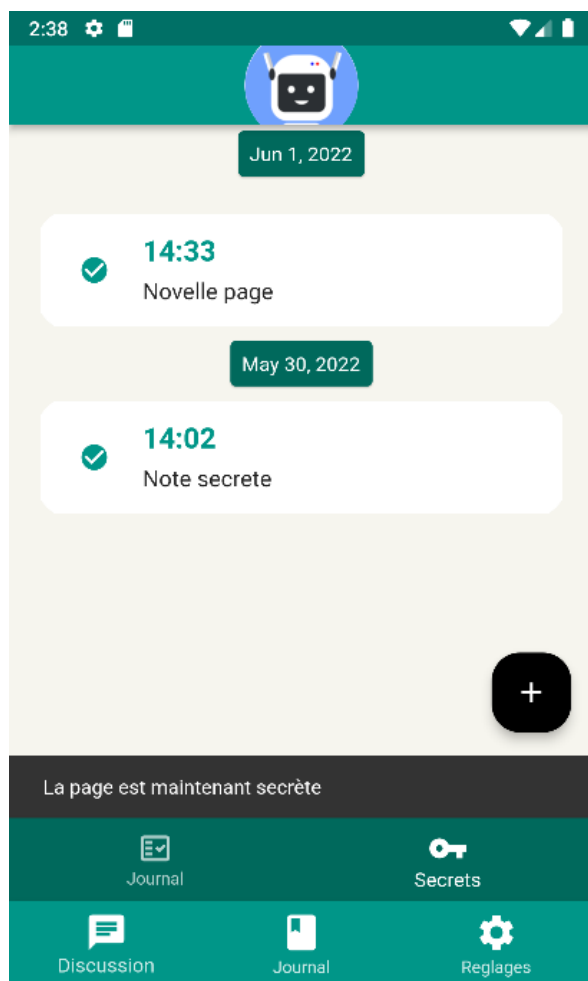
Annexe 9 : Nouvelle page ajoutée



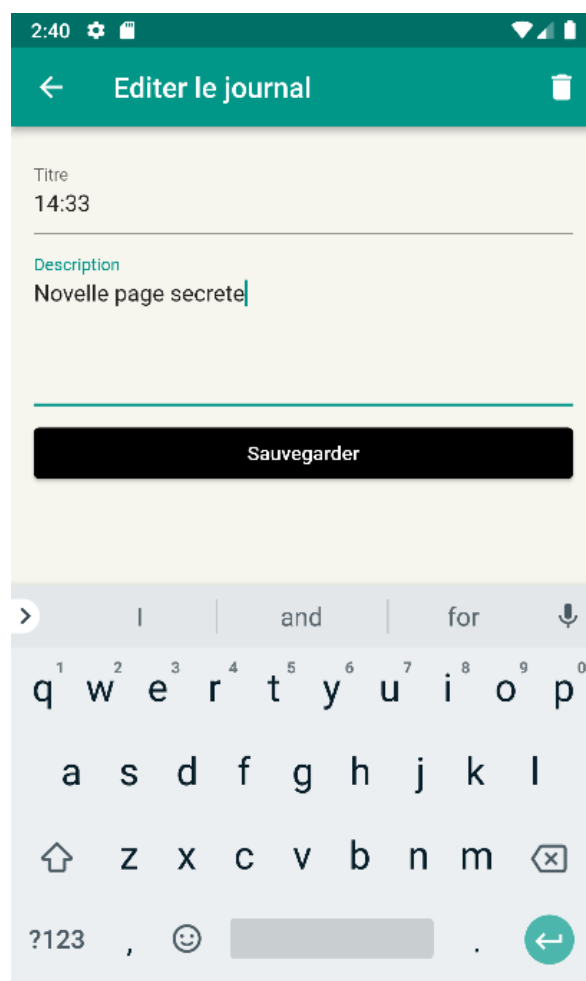
Annexe 10 : Pages secrètes



Annexe 11 : Suppression / modification



Annexe 12 : Page des options



Annexe 13 : Page des options modifiées

3:15

Paramètres




Nom
Pierre

Prénom
Jean

Date de naissance (JJ-MM-AAAA)
01-06-2022

Heure des notifications

Langue

Valider

[Importer](#) [Exporter](#)

Discussion Journal **Reglages**

Annexe 14 : Nouvelle variable %%name

3:16

Paramètres




Nom
Pierre 2

Prénom
Jean 2

Date de naissance (JJ-MM-AAAA)
02-06-2022

Heure des notifications
12.00

Langue

Valider

[Importer](#) [Exporter](#)

Enregistrements effectués

Discussion Journal **Reglages**

Annexe 15 : Journal en Japonais



Annexe 16 : Clear storage



Annexe 17 : Premier démarrage

