

20/03/2024

Projet Mobile

Rapport d'analyse

BRICH Johan

D'HOEST Gage

HENALLUX - IESN

1 TABLE DES MATIERES

2	Attentes du client	2
3	Choix technologiques	3
3.1	Web, pwa, natif ou hybride/cross-plateforme ?.....	3
3.2	Tableau comparatif des technologies	4
3.3	React native	4
4	Analyse Fonctionnelle	5
4.1	Diagramme Use Case	5
4.2	User	5
4.3	Scanner un QR Code	5
4.4	Sélectionner une Épreuve	5
4.5	Démarrer une épreuve	5
4.6	Obtenir un indice	5
4.7	Accès aux technologies/fonctionnalités (sensors).....	5
5	Maquettes	6
5.1	Écran scanner (QR Code)	6
5.2	Écran Carrousel	6
5.3	Écran épreuves.....	7
5.4	Écran indice	7
6	Analyse Technique.....	8
6.1	Diagramme structure de l'application	8
6.2	Diagramme d'activité générale	9
6.3	Diagramme d'activité épreuve 1.....	10
6.4	Diagramme d'activité épreuve 2.....	11
6.5	Diagramme d'activité épreuve 3.....	12
6.6	Diagramme de séquence	13
6.6.1	Utilisateur essaye de démarrer une épreuve	13
6.6.2	Utilisateur complète une épreuve.....	13
7	Sources	15

Rapport d'analyse

Projet Mobile

2 ATTENTES DU CLIENT

- Pas de contraintes technologiques mais évolution possible vers iOS
- QR code pour lancer l'épreuve mais accès aux épreuves débloqué uniquement
- 3 épreuves :
 - L'utilisateur doit pouvoir utiliser la lampe torche pour passer à l'épreuve suivante
 - Gyroscope ou accéléromètre pour charger le canon
 - Mécanisme de notre choix avec l'utilisation de l'une des fonctionnalités du téléphone
- Documentation du projet :
 - Contexte
 - Cahier des charges établi avec le client
 - L'analyse fonctionnelle et technique
 - Justification complète et précise des choix technologiques
 - D'autres points qui nous semblent utiles quand le client reprendra le projet

3 CHOIX TECHNOLOGIQUES

3.1 WEB, PWA, NATIF OU HYBRIDE/CROSS-PLATEFORME ?

- Nous avons choisi une technologie hybride :
 - L'application native est développée sur base de la technologie de l'OS. Elle peut accéder à toutes les ressources (rendant généralement ce type d'application plus fluide) et fonctionnalités du smartphone, mais l'application n'est pas portable sur d'autre OS que celui sur lequel elle a été développée. De plus l'application native est plus lente et coûteuse à développer que ces concurrents développés à l'aide de technologies web ou hybride.
 - Le web lui, permet développer une application à l'aide de technologies web et est accessible directement depuis un navigateur, rendant l'application accessible par n'importe quel device pouvant accéder à un navigateur web. Cela permet, en autres, de ne pas avoir à installer une application depuis un store et économiser de la mémoire de stockage. De plus le développement comporte moins de coûts. Cependant, il est souvent difficile d'avoir accès aux différentes fonctionnalités et capteurs du téléphone ce qui peut limiter les fonctionnalités de l'application (Web API). On peut noter aussi un manque de fluidité à contrario de l'application native.
 - Le PWA se rapproche d'une application web, à la différence qu'elle peut tourner en mode offline, peut être installée comme une application native et est accessible depuis une page web. Cependant elle ne permet pas d'être cross-plateforme.
 - L'application hybride est une application qui va être développer avec une certaine technologie ou langage (par exemple, technologies web pour React-native ou Dart pour Flutter) mais le code lui, sera reconvertis dans langage natif propre à l'OS du téléphone. Cela rend l'application cross-plateforme et permet aussi d'avoir plus d'accès aux fonctionnalités et capteurs du device contrairement à un PWA ou une application WEB, même si l'accès aux ressources n'est pas aussi complète qu'une application purement native. Et à l'instar des application web, l'application peut être moins fluide.

Sur base de ces critères nous avons décidé de choisir une application hybride. En effet, nous avons besoin d'une application qui soit cross-plateforme et avoir accès à un maximum de capteurs du téléphone pour développer les épreuves. L'application hybride était le plus proche de convenir à ces besoins.

3.2 TABLEAU COMPARATIF DES TECHNOLOGIES

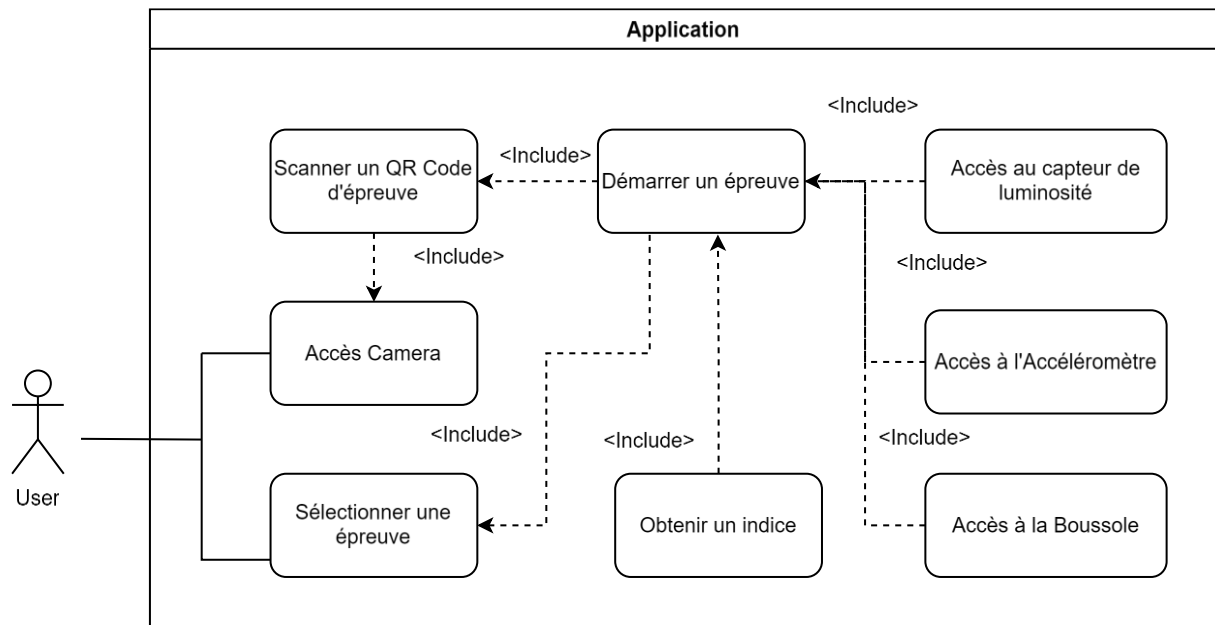
Critères	React-native	Flutter	Windev mobile	Niveau d'importance
Prix	Gratuit	Gratuit	Payant	2
Open source	Oui (+2)	Oui (+2)	Non	2
Courbe d'apprentissage	Faible (+1)	Plus élevé	Plus élevé	1
Rapidité de développement	Bonne (+1)	Bonne (+1)	Très bonne (+2)	2
Connaissance du langage	Oui (+1)	Non	Non	1
Communauté	Très grande (+2)	Plutôt grande (+1)	Pas de comparaison	2
Accès à l'Etat de la lampe torche	Compliqué (+1)	Oui (+3)	Oui (+3)	3
Accès à l'accéléromètre	Oui (+3)	Oui (+3)	Oui (+3)	3
Accès au gyroscope	Oui (+3)	Oui (+3)	Oui (+3)	3
Accès au magnétomètre	Oui (+1)	Oui (+1)	Oui (+1)	1
Accès au capteur de luminosité	Oui (+1)	Oui (+1)	Oui (+1)	1
Accès à la caméra	Oui (+3)	Oui (+3)	Oui (+3)	3
Déployable sur iOS	Oui (+2)	Oui (+2)	Oui (+2)	2
Composants UI	Grande variété (surtout tiers) (+1)	Grande variété native et tiers (+1)	Grande variété native (+1)	1
Score	22	21	19	

3.3 REACT NATIVE

Nous avons choisi React Native car il est gratuit et open-source et que la courbe d'apprentissage est assez faible, mais aussi parce que nous avons la plupart des fonctionnalités nécessaires pour les besoins fonctionnels de l'application. Mais ce qui a fait pencher la balance est le fait que nous avons une connaissance du langage JavaScript ainsi que du framework React par des projets réalisés auparavant. Cependant, nous devons trouver une alternative à la lampe torche pour notre première épreuve, car pour le moment, la librairie Expo ne permet pas d'y avoir accès contrairement à Flutter ou Windev.

4 ANALYSE FONCTIONNELLE

4.1 DIAGRAMME USE CASE



4.2 USER

Toute personne utilisant l'application est considérée comme un utilisateur.

4.3 SCANNER UN QR CODE

Si l'application a accès à la caméra l'utilisateur pourra scanner un QR Code et ainsi démarrer une épreuve.

4.4 SÉLECTIONNER UNE ÉPREUVE

L'utilisateur peut également sélectionner et lancer manuellement une épreuve débloquée via un carrousel.

4.5 DÉMARRER UNE ÉPREUVE

L'utilisateur ne peut sélectionner et ne démarrer qu'une épreuve débloquée.

4.6 OBTENIR UN INDICE

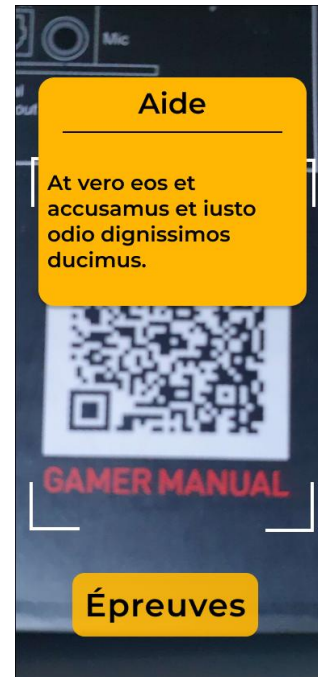
Pour chaque épreuve, l'utilisateur peut demander et obtenir un indice.

4.7 ACCÈS AUX TECHNOLOGIES/FONCTIONNALITÉS (SENSORS)

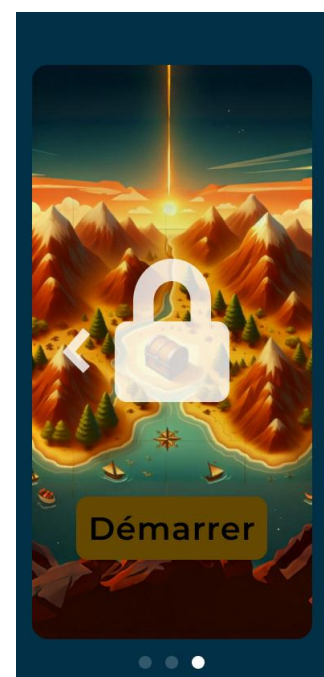
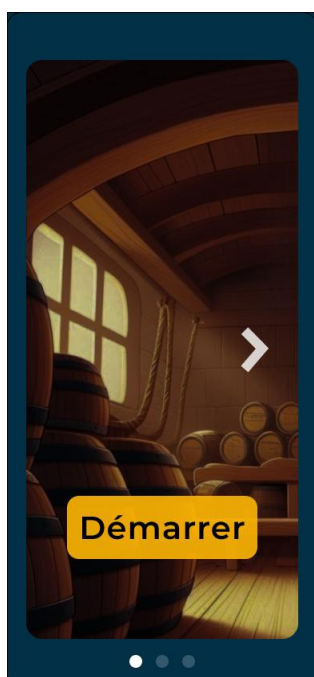
Pour chaque épreuves l'utilisateur pour utiliser les *sensors* de son téléphone (capteur de luminosité, accéléromètre, boussole).

5 MAQUETTES

5.1 ÉCRAN SCANNER (QR CODE)



5.2 ÉCRAN CARROUSEL



5.3 ÉCRAN ÉPREUVES

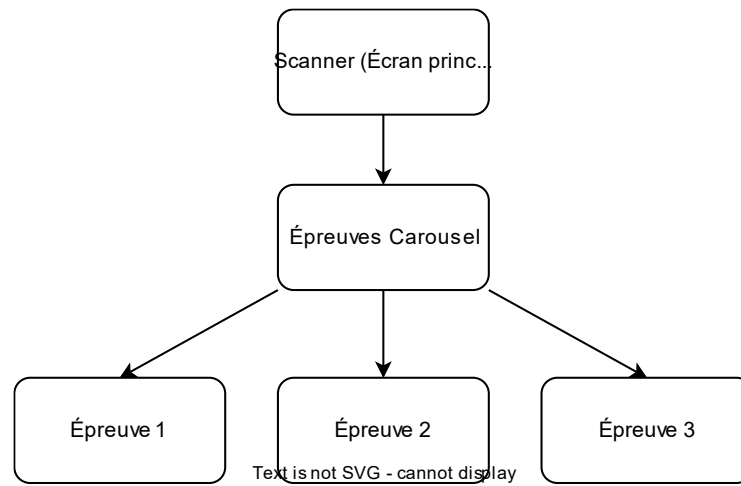


5.4 ÉCRAN INDICE

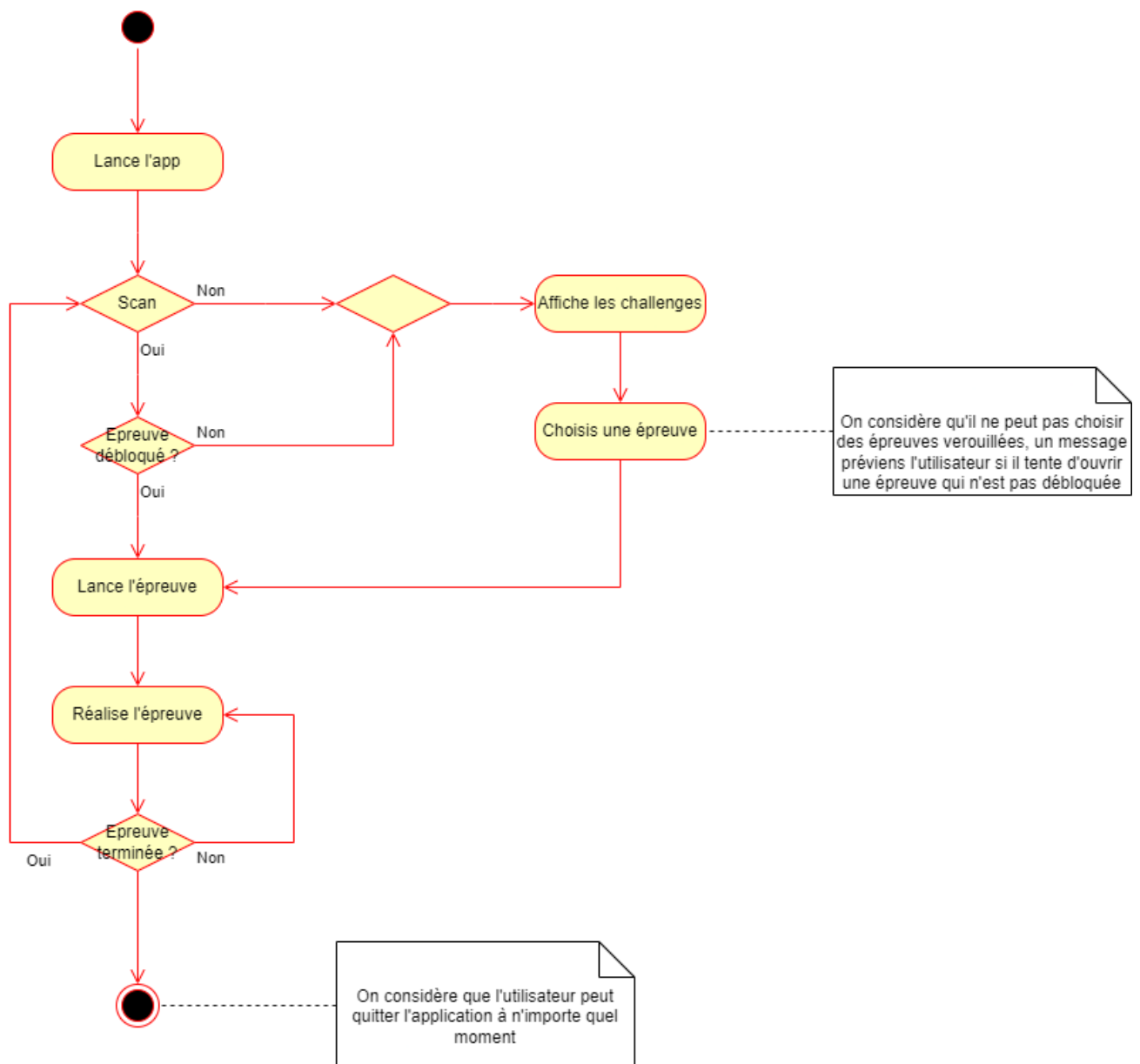


6 ANALYSE TECHNIQUE

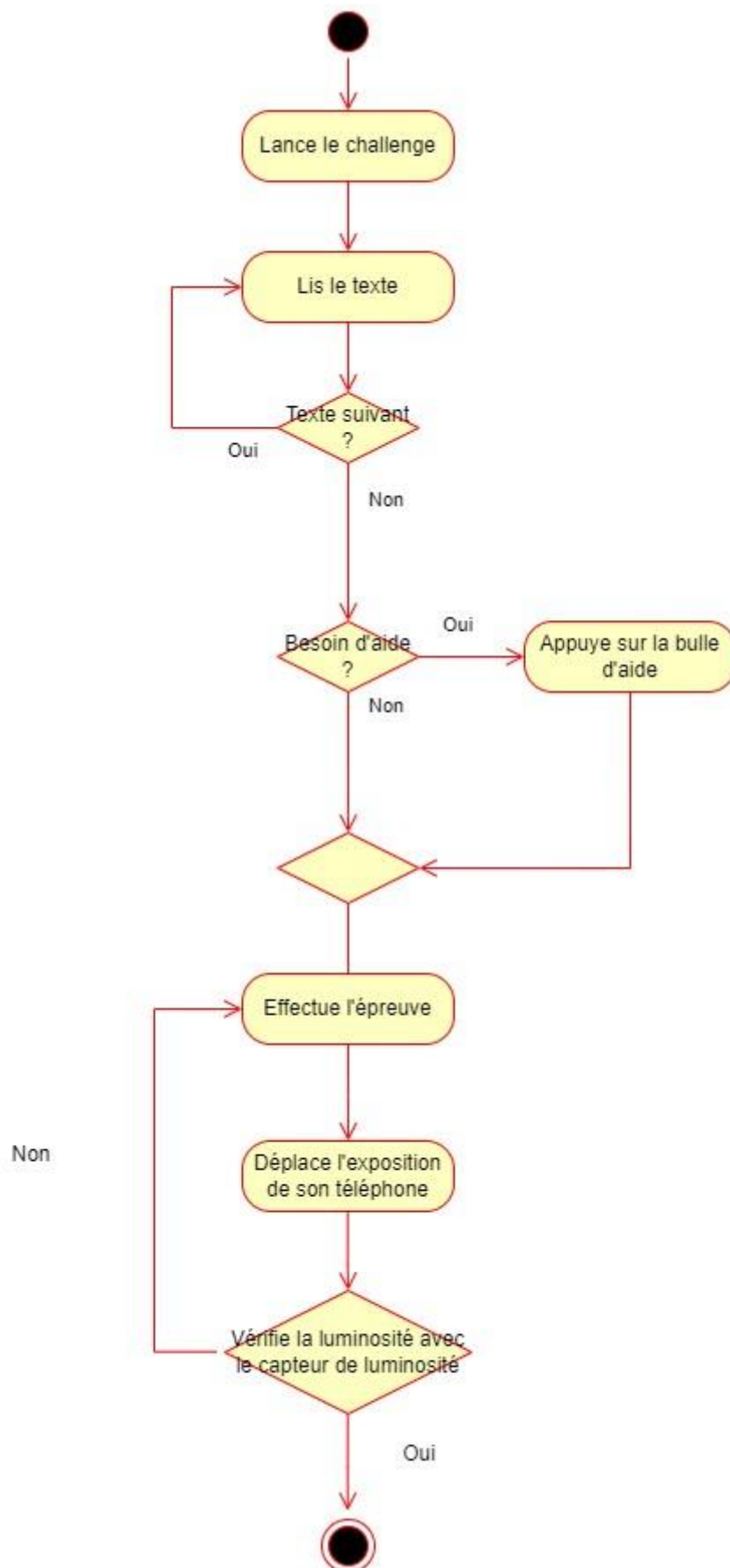
6.1 DIAGRAMME STRUCTURE DE L'APPLICATION



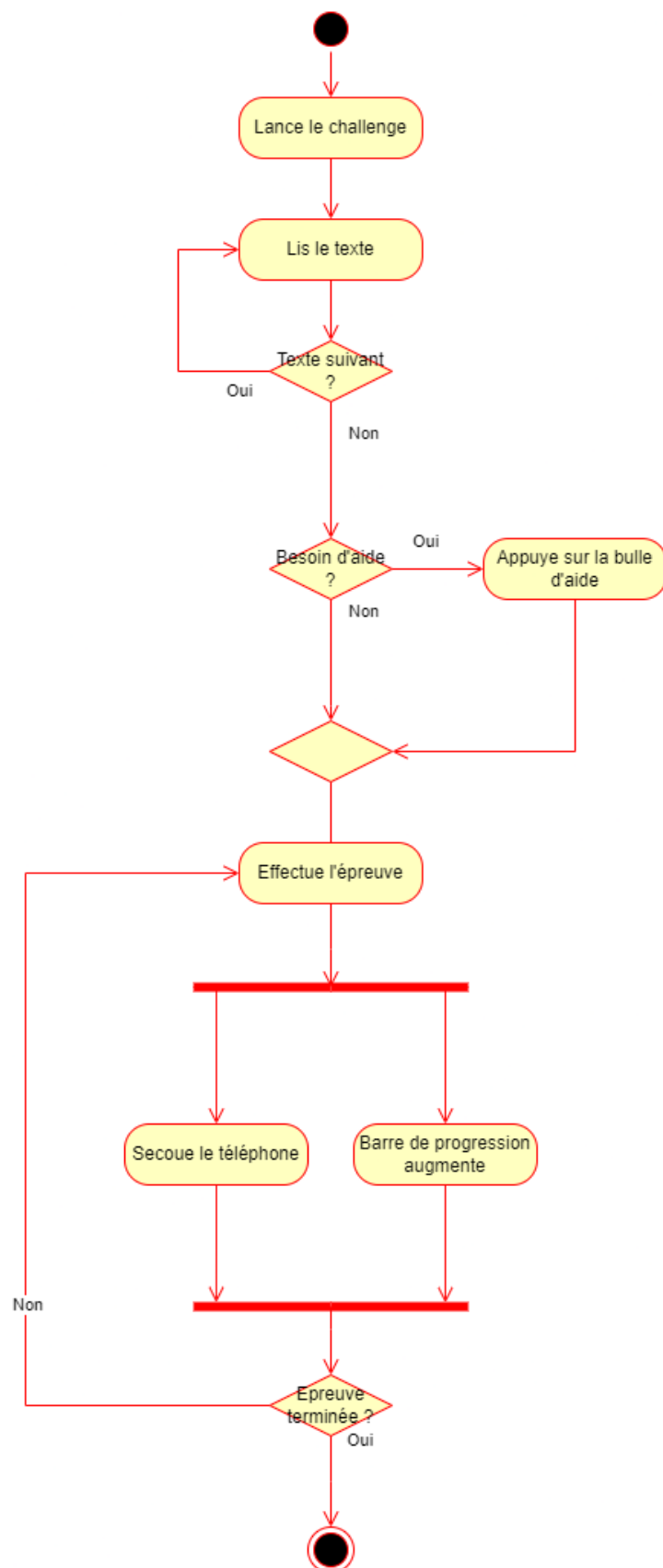
6.2 DIAGRAMME D'ACTIVITÉ GÉNÉRALE



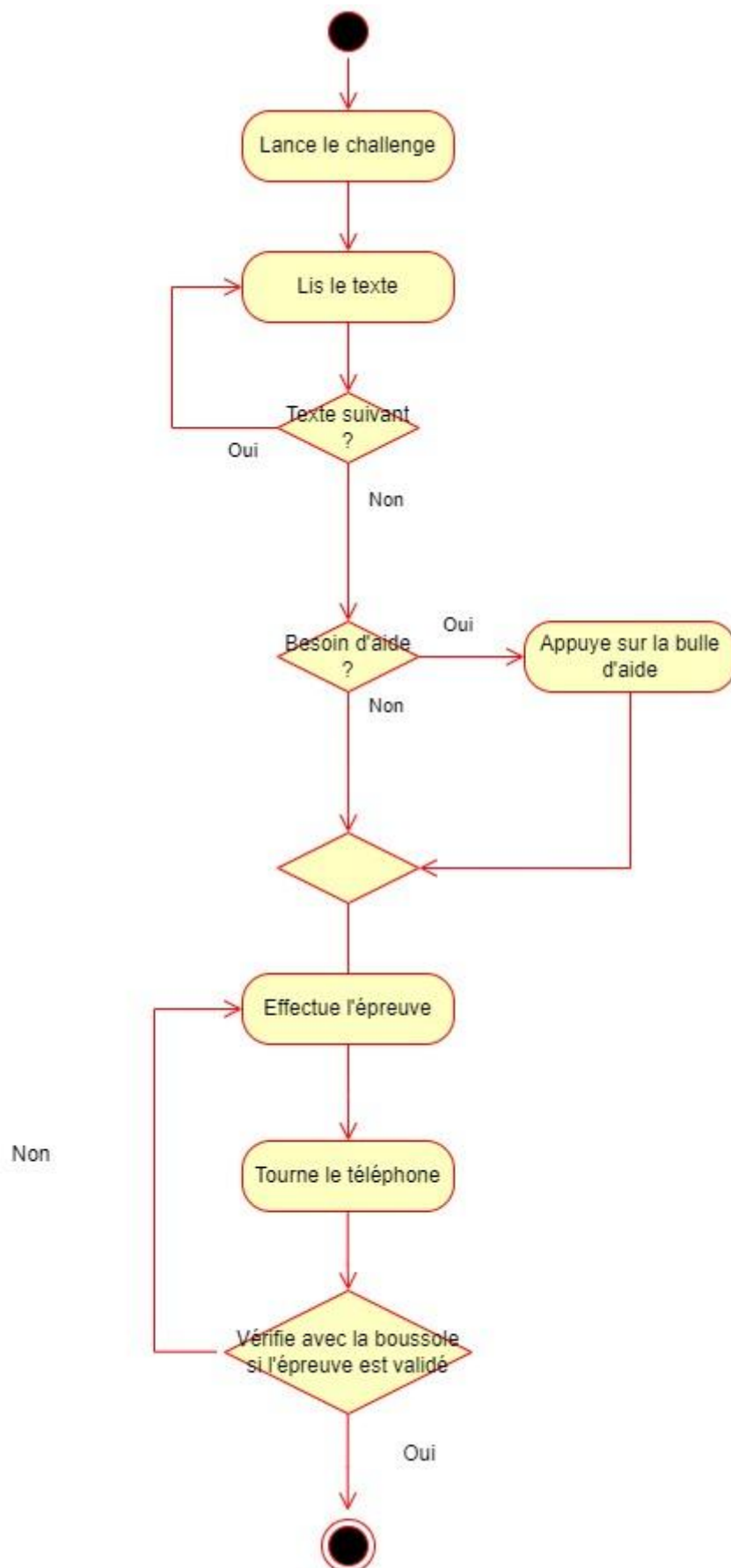
6.3 DIAGRAMME D'ACTIVITÉ ÉPREUVE 1



6.4 DIAGRAMME D'ACTIVITÉ ÉPREUVE 2

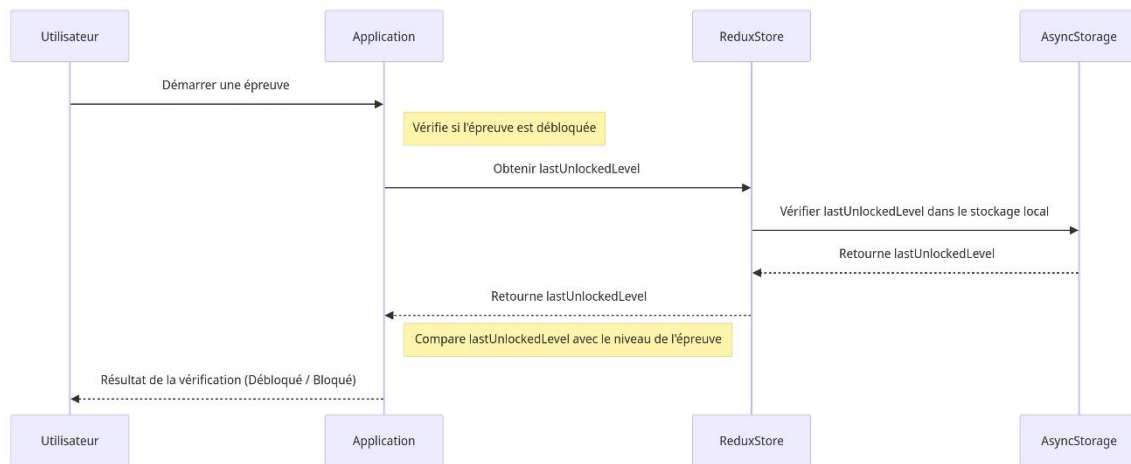


6.5 DIAGRAMME D'ACTIVITÉ ÉPREUVE 3



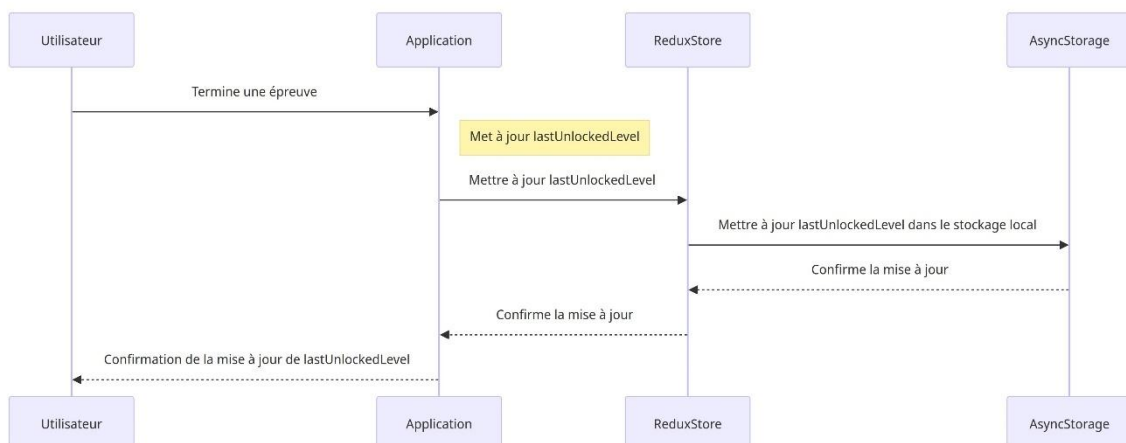
6.6 DIAGRAMME DE SÉQUENCE

6.6.1 Utilisateur essaye de démarrer une épreuve



Quand l'utilisateur veut démarrer une épreuve, l'Application demande lastUnlockedLevel au ReduxStore, qui à son tour vérifie cette valeur dans AsyncStorage. Une fois que AsyncStorage retourne lastUnlockedLevel au ReduxStore, celui-ci le renvoie à l'Application. L'Application compare ensuite lastUnlockedLevel avec le niveau de l'épreuve et renvoie le résultat de la vérification à l'Utilisateur.

6.6.2 Utilisateur complète une épreuve



L'utilisateur a terminé une épreuve. L'Application met ensuite à jour lastUnlockedLevel dans le ReduxStore, qui à son tour met à jour cette valeur dans AsyncStorage. Une fois que AsyncStorage confirme la mise à jour à ReduxStore, celui-ci confirme à son tour la mise à jour à l'Application. Enfin, l'Application confirme à l'Utilisateur que lastUnlockedLevel a été mis à jour.

7 LIENS PROJET

Gitlab du projet : https://gitlab.com/DTM-Henallux/MASI/etudiants/d-hoest-gage/mobile/mobile_project

Clone avec SSH : git@gitlab.com:DTM-Henallux/MASI/etudiants/d-hoest-gage/mobile/mobile_project.git

8 SOURCES

- <https://www.sortlist.fr/blog/pwa/>
- <https://appmaster.io/fr/blog/applications-web-progressives-pwa-et-applications-natives-quel-type-est-le-meilleur-en-2022>
- <https://learn.microsoft.com/fr-fr/microsoft-edge/progressive-web-apps-chromium/>
- <https://radixweb.com/blog/flutter-vs-react-native>
- <https://reactnative.dev/docs/getting-started>
- <https://docs.expo.dev/>
- <https://doc.pcsoft.fr/>
- <https://docs.flutter.dev/>