

---

# Helix

**Application de suivi de performance sportive**

BELLA Jean-Paul

---



# Helix

---

Helix	1
Application de suivi de performance sportive	1
Introduction	3
Framework et technologies utilisés	4
Déroulement du projet	5
Présentation de l'application	7
Quelques problèmes rencontrés	14
Conclusion	15
Références	16

---

## Introduction

La callisthénie correspond à l'ensemble des exercices de gymnastique ou de sport visant à un développement harmonieux du corps. Avec le développement des réseaux sociaux, l'apparence physique et le partage de la vie sociale sont placés au coeur de la société. Il n'est donc pas surprenant de voir se développer des sports tels que la musculation en salle ou le street workout ayant pour but l'esthétique visuelle ou encore la performance, que celle-ci soit absolue en soulevant des charges plus importantes ou bien "visuelle" avec la réalisation de figures nécessitant beaucoup de force.

Ainsi, on dénombre de plus en plus d'abonnés aux salles de musculation et d'applications de suivi de performance et de construction de programmes adaptés aux objectifs de chaque individu telles que Madbarz, Thenics ou Freeletics. Ces applications ont été mes principales sources d'inspiration pour la création d'une nouvelle application dans ce domaine: Helix.

Helix a pour objectif de réunir les avantages de ces différentes applications et d'apporter une touche d'originalité à ce concept. Il s'agit d'une application destinée à toute personne intéressée par le monde de la callisthénie au travers du street workout et/ou de la musculation en salle. Adaptative, Helix permet à l'utilisateur de suivre un objectif choisi parmi le gain en puissance, en volume ou en définition des muscles et d'apporter soutien, suivi et conseils pour la poursuite de cet objectif. Elle permet de surmonter l'obstacle de l'organisation d'un plan d'entraînement personnalisé pour donner l'opportunité à l'utilisateur de se concentrer uniquement sur la réalisation de l'entraînement en maximisant ses chances d'observer des progrès au fil des séances.

---

# Framework et technologies utilisés

## React Native

### Motivations

Le React Native a été le framework retenu pour ce projet. Ce choix s'est basé principalement sur ma volonté d'apprendre ce framework prenant de plus en plus d'ampleur dans le monde de l'auto-entrepreneuriat et du freelance. Ce qui n'était qu'un projet d'un hackathon interne à Facebook en été 2013 se positionne à présent parmi les framework les plus populaires. Désireux d'en connaître les raisons, j'ai alors entrepris mon projet personnel d'application sur cette plateforme.

### Avantages

L'intérêt principal du React Native réside dans ses performances presque impossible à différencier avec celles d'applications simples natives en Java ou en Objectif C tout en permettant une solution multi-plateforme baissant alors le coup d'un développement mobile de manière significative.

Basé sur le JSX (JavaScript-XML), le React Native est donc très facilement accessible étant donné que le JavaScript est un langage de programmation très largement répandu avec très peu de contraintes syntaxiques.

La notion de classe de composants permet la réutilisation efficace du code qui, couplée à la présence d'une communauté maintenant développée, donne accès à de nombreux composants "prêts à l'utilisation".

### Inconvénients

La première chose qui m'a frappée lors de mes débuts est l'absence d'un module unifiant la navigation de pages en pages sur iOS et sur Android sans avoir recours à des configurations propres à chacune des plateformes. Or, ce processus de séparation du code, amène de nombreuses complications difficiles à gérer pour un novice. Il m'est donc paru préférable pour une première application en React Native de profiter un maximum de la plateforme sans cette séparation. En effet, cette opération de séparation appelée "exp detach" bloque la possibilité de publier son application en une commande sur la plateforme expo, la capacité de "tunneling" et toutes les API d'Expo. Je me suis ainsi contenté dans un premier temps de l'exploitation au maximum des ressources mises à ma disposition pour envisager par la suite une séparation.

---

Malgré l'utilisation de ce framework depuis maintenant plus de cinq ans, la plateforme est toujours en position d'instabilité au niveau d'API plus poussées telles que la reconnaissance vocale ou la géolocalisation qui ne sont toujours pas disponibles même au travers d'Expo. De plus, les composants "tout fait" sont encore très peu présents et on doit souvent parcourir la communauté aux travers de nombreux GitHub pour trouver un composant intéressant et l'adapter alors à son utilisation car la documentation officielle n'en présente pas autant.

Les performances, bien qu'identiques entre deux applications simples faites en React Native ou en langage véritablement natif s'éloigne lorsque de nombreux composants tels que des cartes, des animations lourdes ou s'adaptant aux différentes vues utilisateurs interviennent. Le React Native semble ainsi s'adapter davantage pour l'instant à développer des applications simples et de manière rapide en évitant un nombre de couches trop importantes comme ses concurrents hybrides tels que Cordova.

## Expo

Expo est un ensemble d'outils, de bibliothèques et de services permettant de construire une application native iOS et Android en écrivant du JavaScript.

L'application Helix est à l'heure actuelle une "Expo App" puisqu'elle est une application en React Native contenant l'Expo SDK qui est une librairie JavaScript native permettant d'accéder aux fonctionnalités du système du téléphone mobile.

## Déroulement du projet

Un travail préliminaire de préparation a été nécessaire afin de se former au React Native. J'ai ainsi débuté par la lecture du court tutoriel proposé sur la documentation officielle. Après cette sensibilisation au langage, je me suis immédiatement lancé dans la réalisation de croquis de l'application et sur la conception des différentes pages. En effet, mon projet étant conséquent, je devais démarrer assez rapidement pour connaître par moi-même les difficultés dans l'implémentation de code en React Native et me faire très vite une idée de la réelle difficulté du Framework.

Bien que le React Native repose sur un langage simple, le JavaScript, la compréhension de la structure du projet est bien différente de celle existante en Java. N'ayant pas d'expérience en Web auparavant, l'utilisation du CSS et du système de flex box n'a pas été un avantage pour moi au début. Cependant, vers le milieu du projet, je me sentais davantage à l'aise et j'ai pu constater un net gain de temps à l'utilisation du CSS et non du XML pour le design des pages. Le lancement du projet s'est donc fait relativement rapidement afin de rencontrer le maximum de difficultés possibles pour pouvoir cibler les notions à retravailler et à revoir par la suite. Un grand nombre d'heures a par conséquent été réalisé à ce moment là. Mon professeur encadrant le projet, Monsieur Sylvain

---

GIROUX, avait pour rôle de superviser à distance mon projet. Cependant, de par la distance nous séparant et ma volonté d'autonomie, je préférais choisir l'utilisation des forums tels que StackOverflow ou encore OpenClassRoom et ne garder les contacts que pour des compte-rendus de l'avancement du projet de façon hebdomadaire.

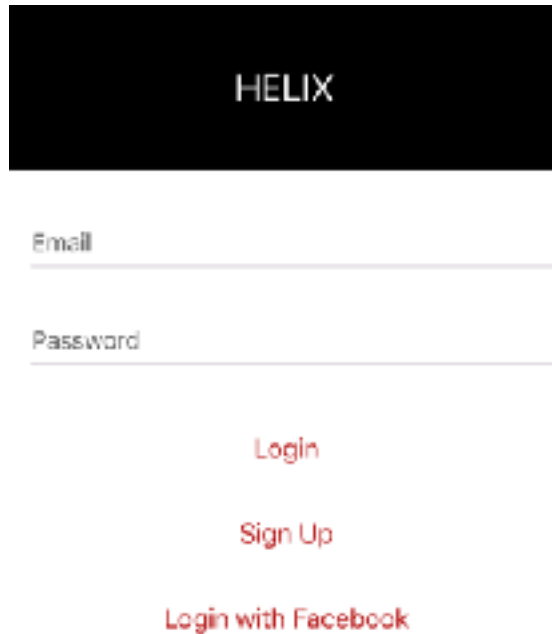
Le plan de projet a été en grande partie respecté à l'exception du partage de photographies de motivation qui a été simplement mis de côté par manque de temps, la possibilité de réorganiser les exercices à l'aide d'une flatlist ré-ordonnable et l'utilisation de TensorFlow. En effet, la flatlist étant un composant propre de React Native, elle n'est pas facilement modifiable et rend compliqué la gestion du z-index pour donner l'illusion d'une vue par dessus l'autre. Quant à TensorFlow, après étude de son implémentation, son importance a été amoindrie et j'ai finalement décidé en priorité de régler les problèmes d'affichage sur iOS que de forcer son utilisation. TensorFlow est effectivement intéressant dans l'étude d'un groupe d'individus et aurait pu donc être utilisé pour des groupes d'individus à l'objectif commun en supplément des modifications propres à chaque individu. Cependant, il aurait fallu procéder au détachement de l'Expokit pour cette unique fonctionnalité, qui en plus d'être difficile à tester car ne disposant pas actuellement de nombreuses personnes utilisant l'application, aurait posé des problèmes de concurrence avec les changements propres à l'individu.

Toutefois, j'ai été surpris de voir la quantité de travail personnel demandé pour ce projet. En France, nous ne disposons en effet en moyenne que d'un ou deux projets de cette ampleur par session laissant donc facilement du temps à consacrer pour ce travail personnel. Cependant, la présence de multiples projets semblables dans les différentes matières ne m'a pas laissé suffisamment de temps à consacrer à ce projet que j'aurais aimé pousser jusqu'à la création de badges de récompense pour différents challenges sportifs ainsi qu'au détachement de l'ExpoKit.

---

## Présentation de l'application

### Page de connexion

A screenshot of a mobile application's login page. At the top, there is a black rectangular header with the word "HELIX" in white, uppercase letters. Below the header, there are two input fields: the first is labeled "Email" and the second is labeled "Password", both in a light gray font. Each label is positioned to the left of its respective input field, which is a thin, light gray line. Below the input fields, there are three red text links: "Login", "Sign Up", and "Login with Facebook", arranged vertically. The background of the page is white.

HELIX

Email

Password

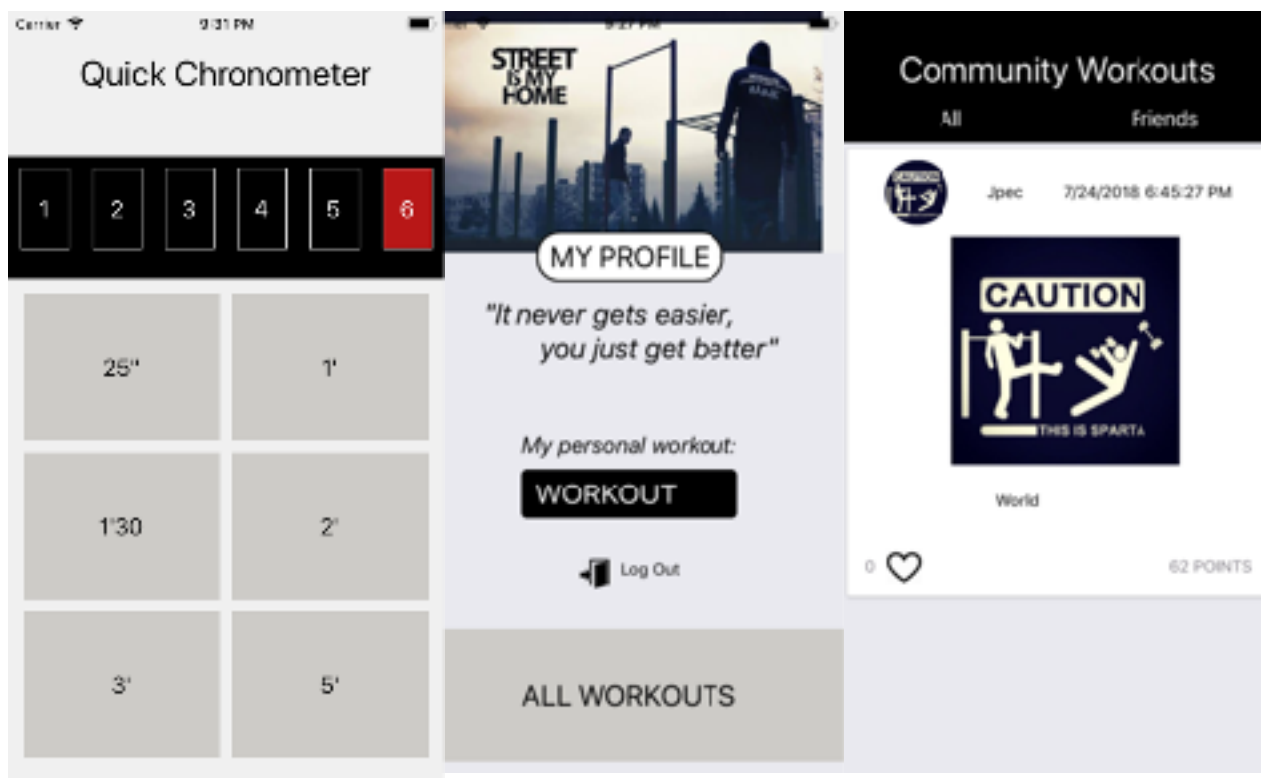
Login

Sign Up

Login with Facebook

Cette page permet de s'enregistrer et/ou de se connecter à l'application. L'utilisateur peut choisir la connexion par email ou par Facebook directement.

## Page d'accueil



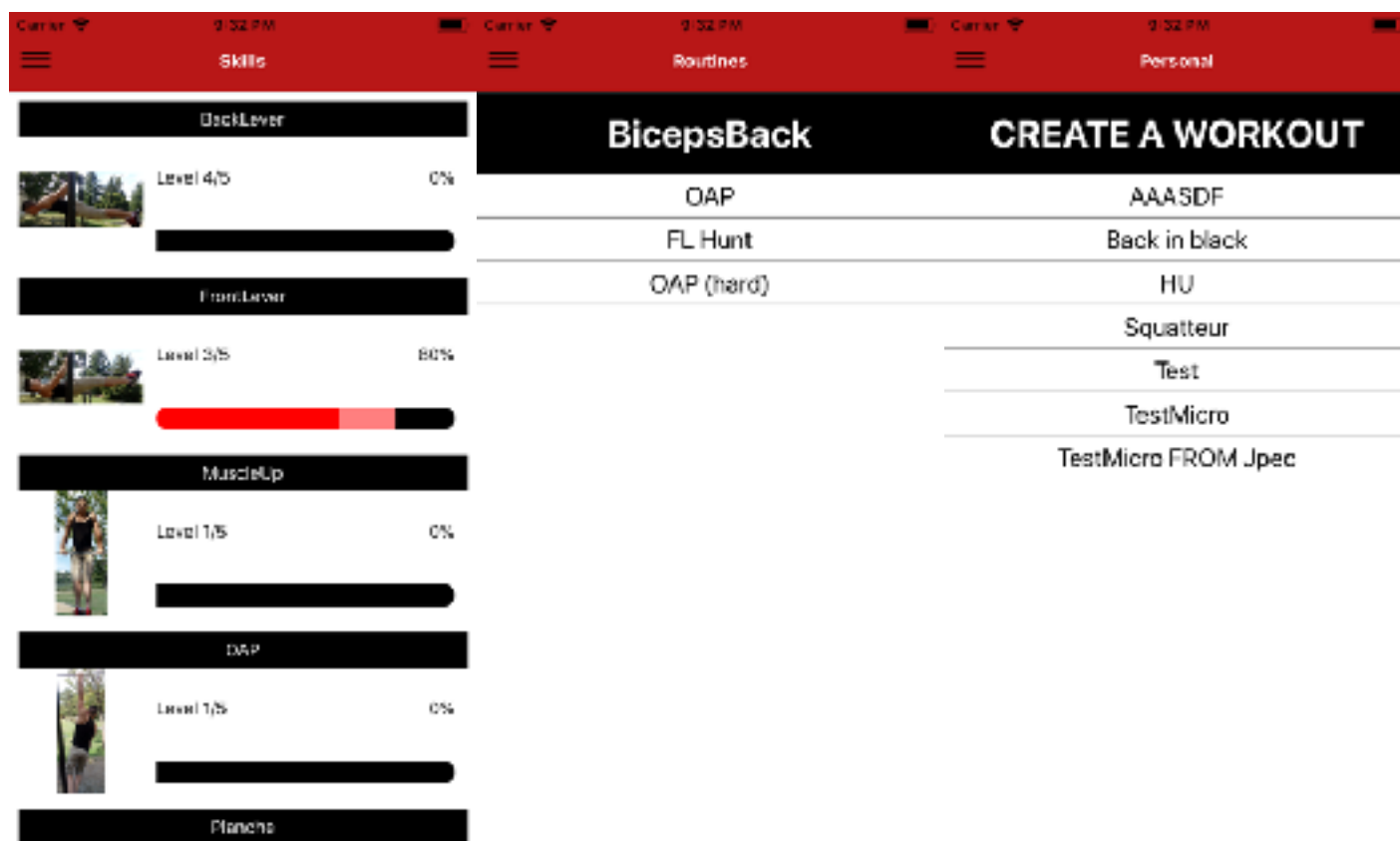
Ici, il est possible d'accéder au profil utilisateur en cliquant sur *"My Profile"*, de lancer l'entraînement personnalisante en cliquant sur *"WORKOUT"*, ou à l'ensemble de la section d'entraînement en cliquant sur *"All workouts"* ou de se déconnecter en cliquant sur l'icône correspondante.

De plus, en glissant le doigt vers la droite, un chronomètre rapide est mis à disposition pour que l'utilisateur puisse aussi réaliser des entraînements sans désirer enregistrer quoique ce soit, mais respecter un certain temps de repos.

En glissant le doigt vers la gauche, on accède à la liste des entraînements réalisés par les membres de l'application que l'on peut filtrer pour n'afficher que ceux de nos amis.



## Page de sélection d'entraînement

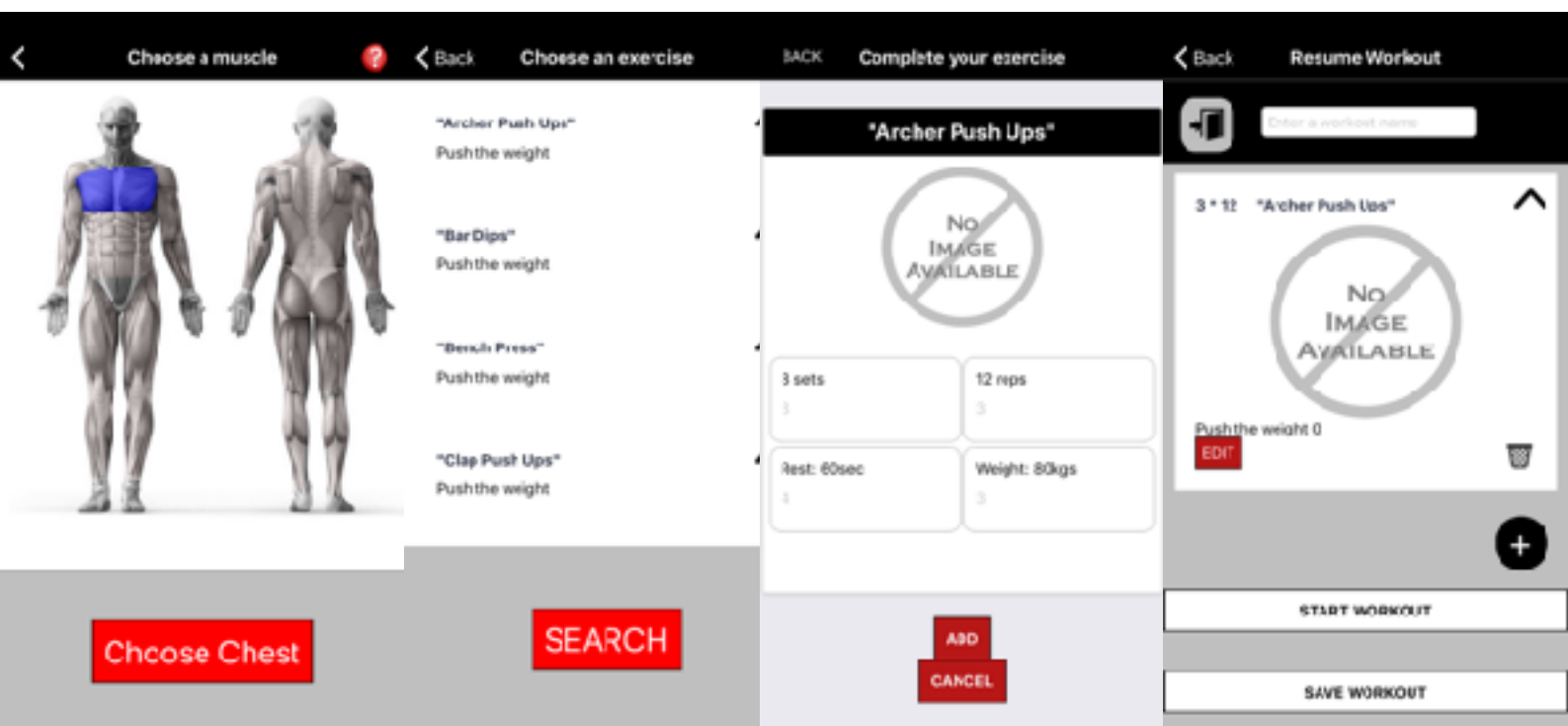


L'utilisateur peut choisir le muscle ou groupe musculaire qu'il compte viser en cliquant sur le filtreur "BicepsBack" faisant apparaître un dialogue proposant les différents choix possibles et il lui suffit alors de cliquer sur un entraînement pour observer son descriptif.

A gauche, l'ensemble des figures (skills) avec l'avancement dans les niveaux de l'utilisateur est disponible. En cliquant sur un *skill* particulier, l'utilisateur disposera de cinq niveaux qu'il doit débloquent progressivement jusqu'à atteindre le *skill* en question.

A droite, l'utilisateur peut choisir un entraînement qu'il a déjà conçu ou en créer un nouveau en cliquant sur "Create a workout".

## Pages correspondant à la création d'un entraînement

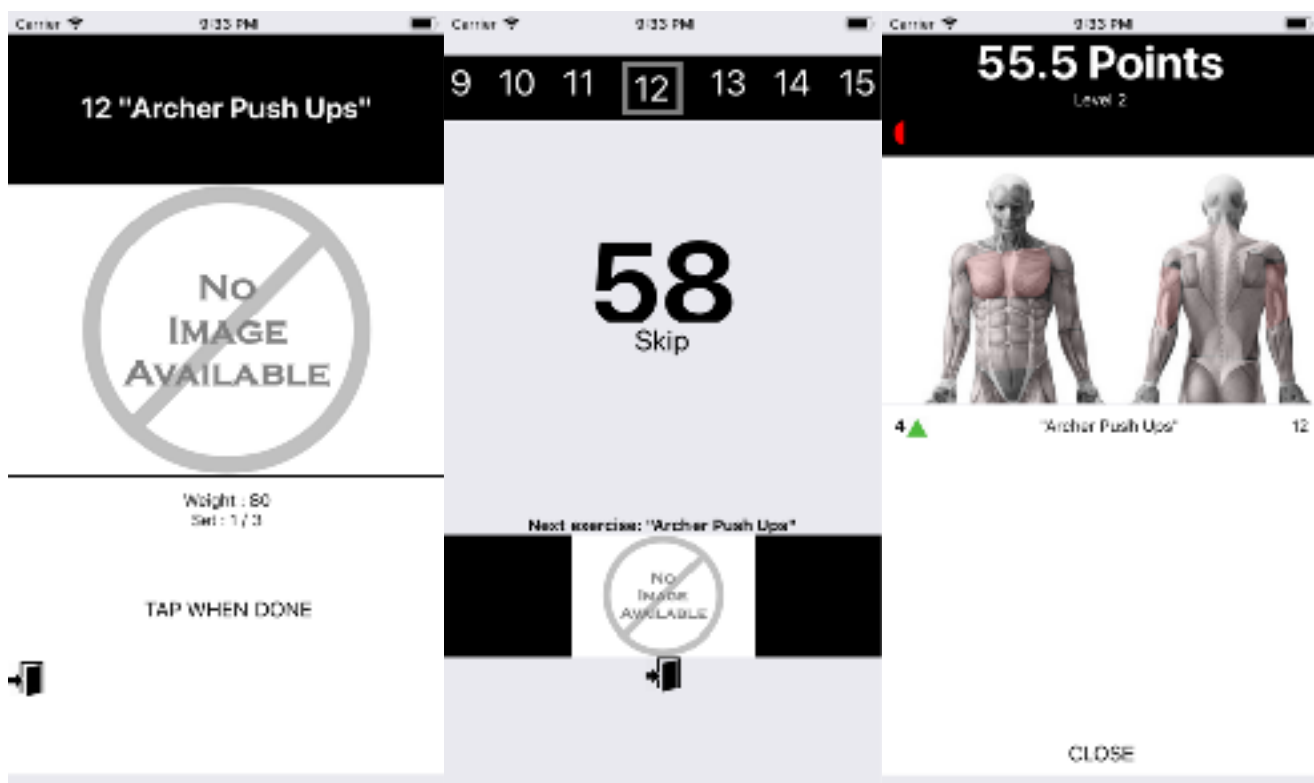


Pour créer un entraînement, l'utilisateur doit:

- Choisir un muscle en cliquant sur le muscle en question
- Cliquer sur l'exercice qui l'intéresse avec la possibilité d'obtenir des renseignements sur l'exercice avec l'aide du descriptif
- Préciser le nombre de répétitions, de séries, de temps de repos, de poids utilisés de l'exercice.

C'est alors qu'il arrive à un récapitulatif montrant l'état actuel de l'entraînement ainsi construit. Il peut décider de l'enregistrer ou de simplement l'exécuter une unique fois.

## Déroulé d'un entraînement



Une fois entré sur un programme de son choix, l'utilisateur aperçoit l'écran de gauche lui indiquant le nombre de répétitions et le nom de l'exercice à effectuer. Il peut, après réalisation de l'exercice lancer le temps de repos en cliquant sur "*Tap when done*", l'amenant alors sur le second écran.

Ici, l'utilisateur peut changer le nombre de répétitions pour indiquer le nombre réellement réalisé en cliquant sur les chiffres pour faire tourner le sélecteur. Il peut à tout moment quitter l'entraînement en cliquant sur l'icône en forme de porte ou bien, il parcourra l'ensemble des exercices de l'entraînement en alternant ces deux vues.

Enfin, l'écran de droite apparaîtra à la fin résumant les exercices réalisés et montrant les zones musculaires travaillées.

## Page de calcul de maximum

Carrier 5:34 PM

<

### 1RM Calculator

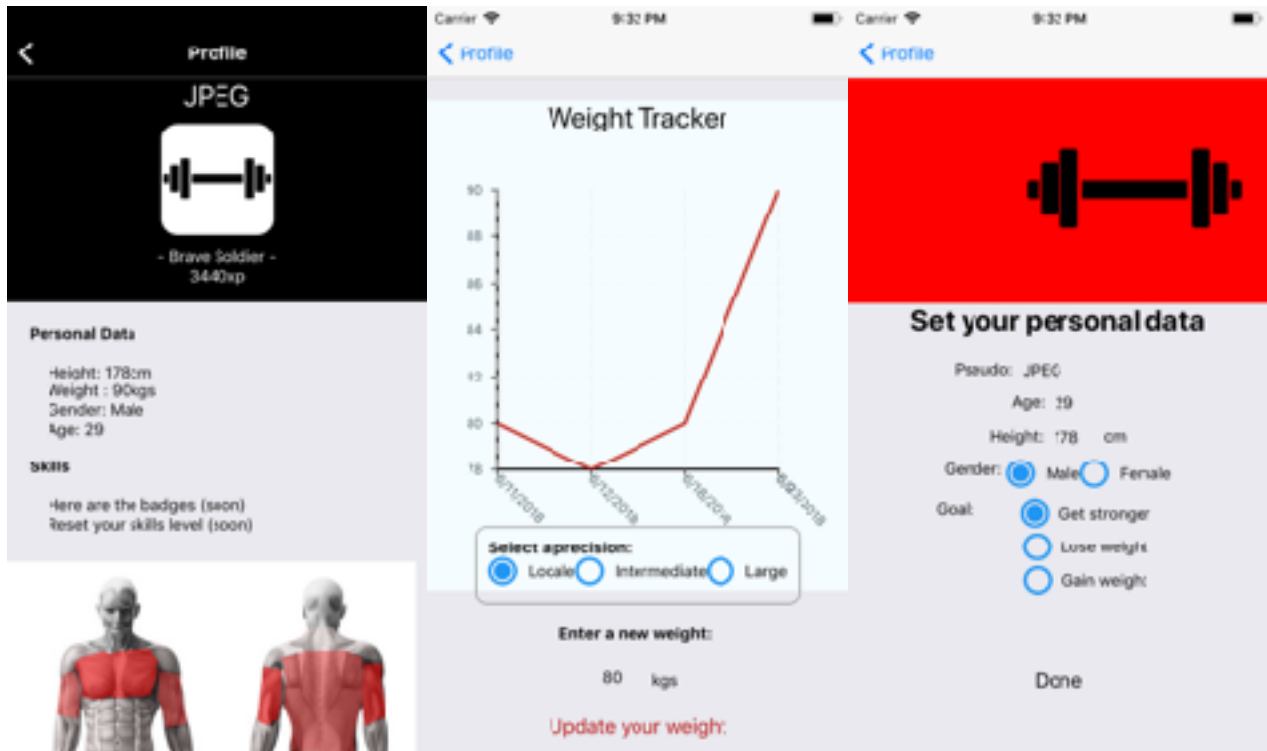
1RM

130

Reps	Weight
1	130.00
2	123.50
3	120.50
4	117.00
5	113.10
6	110.50

Cette page permet à l'utilisateur de connaître le poids maximum qu'il peut espérer soulever et connaître le poids à utiliser pour réaliser un nombre de répétitions donné. Changer un champ met automatiquement à jour les autres.

## Pages correspondant à la gestion du profil utilisateur



Cet ensemble de pages correspond à la gestion actuelle du profil utilisateur. L'utilisateur peut observer les muscles qu'il a principalement travaillé avec un système de colorisation simple: plus la couleur rouge du muscle est prononcée, plus ce muscle a été travaillé. Il a accès à son poids qu'il peut changer et observer alors l'évolution sur un graphique adapté.

---

## Quelques problèmes rencontrés

### Navigation Drawer

Les principales difficultés ont été rencontrées avec la navigation. Il est en effet très difficile avec React Navigation de gérer des routes distinctes pour différents composants. Lors de la réalisation du *Navigation Drawer*, j'ai du apprendre à gérer les contextes en réalisant plusieurs *binding*, mais cela n'a pas suffi à retrouver les routes qui devenaient alors inaccessibles après une redirection faite par le *Navigation Drawer*. En effet, une fois la redirection faite, l'application ne conservait en mémoire que les routes contenues dans le *Navigation Drawer* sans possibilité de réinitialisation. Afin de pallier à ceci, j'ai donc décidé de créer simplement une vue dynamique se superposant aux autres avec un z-index plus important. Une autre vue venait remplir l'écran de manière transparente pour gérer le clic hors de la vue principale pour permettre la disparition de la vue principale.

### Drag and Drop et liste triable

La difficulté provenant de la liste triable venait de l'utilisation d'un composant immuable de la documentation: la *flatlist*. Seules certaines propriétés choisies pouvaient alors être accédées. Il aurait fallu dans le cas présent réalisé un composant se répétant sur l'ensemble des items d'une liste. Bien que la performance en serait alors réduite, il aurait été possible de recréer ce composant pour y appliquer un *PanResponder* pour gérer les mouvement au sein de cette même unité contrairement au bloc unité n'étant que des lignes dans les *flatlists*. Je n'ai toutefois pas eu le temps de tenter l'implémentation de cette solution technique.

---

## Conclusion

Malgré une documentation assez succincte, la présence d'une communauté active et dynamique rend le React Native très intéressant pour le développement d'applications mobiles. Le développeur doit faire un premier effort pour bien saisir les différents concepts simples expliqués dans la documentation et découvrir de lui même les subtilités en s'attaquant à des problèmes nécessitant des API plus complexes. Ceci est à la fois la force et la faiblesse du React Native: il s'agit d'un grand saut dans l'inconnu avec quelques bases, mais proposant des possibilités innombrables. Pour le moment, il demeure, à mon simple avis, un langage plus adapté à la mise en développement d'applications simples, mais l'évolution du langage continue à grande vitesse et il ne m'étonnerait pas de voir cette incommodité disparaître dans un futur proche.

J'ai pu ainsi réaliser un projet qui me tenait véritablement à coeur tout en étant satisfait du résultat final, bien qu'encore incomplet. Afin de rendre mon application plus attractive, une amélioration du design général reste encore nécessaire ainsi qu'un rajout de contenu dans les entraînements et exercices proposés. Cependant, ce travail aurait nécessité un nombre beaucoup plus importants d'heures que je compte réaliser à la fin de cette session. Ces modifications auraient pu être ajoutées de façon sommaire, mais je ne préfère mettre en ligne une version de l'application au point sur les pages présentées afin d'éviter une impression de travail bâclé. A cela, j'aurais aimé ajouter un nombre plus conséquent d'animations pour dynamiser l'application, ainsi qu'un système de notifications pour inviter l'utilisateur à s'entraîner si sa dernière connexion est très ancienne ou si l'un de ses amis décide de le notifier.

Je pense toutefois qu'un utilisateur n'ayant aucune connaissance préalable pourrait se servir dès à présent de l'application, plus particulièrement de l'espace *Skills* et *Personal Workout* pour s'améliorer de façon significative.

---

## Références

<https://citrusbits.com/react-native-pros-and-cons/>

<https://uxplanet.org/react-native-vs-cordova-phonegap-ionic-etc-2f85d9651605>

<https://levelup.gitconnected.com/react-native-with-expo-is-a-joyride-671d706b7a19>

<https://medium.com/the-react-native-log/comparing-the-performance-between-native-ios-swift-and-react-native-7b5490d363e2>