

Android Cycling Trainer



Lechaire Thomas – FIN2
ETML - Lausanne
102 heures
M. Patrick Chenaux

Table des matières

1	Spécifications	3
1.1	Spécifications de départ et documentation associée	3
1.1.1	Objectifs et portée du projet	3
1.1.2	Fonctionnalités requises (du point de vue de l'utilisateur)	3
2	Planification	4
2.1	Planning(s) initial	4
3	Analyse	6
3.1	Faisabilité	6
3.1.1	Les compétences à acquérir :	6
3.1.2	Matériel/Logiciels	6
3.1.3	Recherches d'informations/documentation	6
3.1.4	Gestion du temps	6
3.1.5	Choix des logiciels	6
3.2	Document d'analyse et conception	8
3.3	Conception des tests	8
3.4	Planning détaillé	8
4	Réalisation	8
4.1	Dossier de réalisation	8
4.2	Modifications	9
5	Tests	9
5.1	Dossier des tests	9
6	Conclusion	9
6.1	Bilan des fonctionnalités demandées	9
6.2	Bilan de la planification	9
6.3	Bilan personnel	9
7	Divers	9
7.1	Journal de travail de chaque participant	9
7.2	Bibliographie	9
7.3	Webographie	9
8	Annexes	10

1 Spécifications

1.1 Spécifications de départ et documentation associée

1.1.1 Objectifs et portée du projet

- Analyser attentivement le cahier des charges.
- Proposer une interface graphique simple et intuitive.
- Réaliser l'application sous Android.
- Tester l'application.

1.1.2 Fonctionnalités requises (du point de vue de l'utilisateur)

L'utilisateur doit pouvoir ...:

- Créer de façon simple une séance d'entraînement par exemple :
 - Entraînement route « force » : 6 sprints 8''/52'' + 6 sprints 12''/48'' + 5 x 1' + 3 x 3' + 1 x 5' + 3 x 3' + 6 sprints 10''/50''.
 - Entraînement VTT « côtes » : 2 tours de 3 côtes de 5' une doublée (7 côtes) : 4^e et 5^e en 15/15 et 6^e et 7^e en 20/20.
 - Entraînement route « rythme » : 30' échauffement avec 10 sprints de 6 à 8'', 10' 20/20 côte 160-165 bpm, 15' vitesse 130 bpm, 3' 30/30 côte à bloc 180 bpm, 15' vitesse 130bpm, 20' contre-la-montre 165 bpm souplesse, 15' vitesse 130 bpm, 20' contre-la-montre 165 bpm braquet + gros, 20' retour au calme.
- Afficher un résumé de la séance d'entraînement, cela permet d'avoir un aperçu global de l'entraînement.
- Démarrer la séquence d'entraînement, chaque partie de la séquence doit être affichée de façon claire, au moyen de couleurs et contrôles graphiques adéquats, barres de défilement, boutons, clignotements etc.
- Mettre en pause la séquence d'entraînement et la redémarrer au moment souhaité.
- Sauvegarder une séance d'entraînement créée.
- Charger une séance d'entraînement précédemment sauvegardée.
- Insérer sa fréquence cardiaque de repos et maximale, elle pourra être utilisée pour travailler ou afficher une certaine zone cardiaque.
- Calculer un indice de récupération sur le vélo pour une date précise : $F_{cmax} - FC$ 1'30'' après effort, cette valeur augmente lorsque la forme physique augmente, afficher l'historique des valeurs enregistrées.

2 Planification

2.1 Planning(s) initial

- Dates de réalisation : du lundi 11.05.2015 au lundi 08.06.2015
- Horaire de travail :
 - Lundi** 08h00-12h15 13h10-15h40 Pentecôte le 25 mai
 - Mardi** 08h00-11h25 13h10-16h35
 - Mercredi** - 13h10-16h35
 - Jeudi** 08h00-11h25 12h20-15h40 Ascension le 14 mai
 - Vendredi** 08h00-11h25 12h20-15h40 Pont de l'Asc. le 15 mai
- Nombres d'heures : 102

Tâches - objectifs	Nb 1/4 heure	Semaine 1
Absence - Imprévus	72	
Planification	0	
Analyse	24	
Documentation	0	
Maquette de réalisation	21	
Programmation	0	
Tests	0	
Debugging	21	
Rédaction Rapport et Journal de travail	0	
Zone Tampon	20	
Total planifié	95	
Total réalisé	0	

2.1 Semaine 1

Tâches - objectifs	Semaine 2
Absence - Imprévus	
Planification	
Analyse	
Documentation	
Maquette de réalisation	
Programmation	
Tests	
Debugging	
Rédaction Rapport et Journal de travail	
Zone Tampon	
Total planifié	
Total réalisé	

2.2 Semaine 2

2.3 Semaine 3

2.4 Semaine 4

2.5 Semaine 5

3 Analyse

3.1 Faisabilité

- Ce paragraphe énumère la liste des difficultés potentielles de tout ordre:
- Dans le cadre de ce projet, une analyse est nécessaire afin de pouvoir anticiper les problèmes qui pourraient survenir durant la réalisation ou la programmation. Il faut vérifier certains points comme :
 - les compétences à acquérir ou approfondir
 - le matériel nécessaire ou à exploiter
 - les recherches d'informations particulières
 - la gestion du temps de travail
 - Sélection des logiciels
 - les difficultés potentielles et les solutions envisagées

3.1.1 Les compétences à acquérir :

- Compétences en java et POO
- Compétences en programmation Android
- Bases en XML
- Compétences en Sql ou Base de données
- Connaissances de l'environnement Android Studio

3.1.2 Matériel/Logiciels

- Un Ordinateur PC
- Microsoft Office
- Android Studio
- Virtual Box
- Gimp ou Inkscape pour le traitement d'image
- Navigation Internet

3.1.3 Recherches d'informations/documentation

- Recherches sur les bases de données Android
- Documentation sur l'environnement Android Studio
- Recherches sur les solutions pour les Tests ou Debugging
- Recherches sur les interfaces Android
- Documentation de Google pour le développement Android

3.1.4 Gestion du temps

La gestion du temps se fait grâce à la planification Initiale ainsi qu'au journal de bord qui permettent de suivre un fil rouge. La gestion du temps peut facilement devenir problématique. Il est donc important de respecter au maximum les délais fixé dans la planification (initiale et détaillée) afin de mener le projet à son terme.

3.1.5 Choix des logiciels

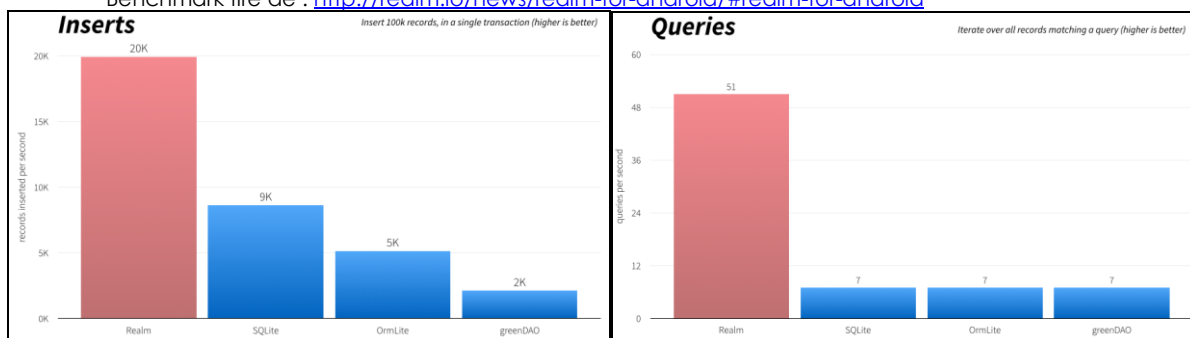
Concernant le logiciel de développement. Deux grands logiciels peuvent être envisagés. Eclipse projet d'Eclipse Foundation qui possède un plugin (ADT) qui s'intègre très facilement à Eclipse et qui permet de développer sous Android, ainsi que Android Studio qui est développé par Google et qui se base sur IntelliJ IDEA (autre logiciel de développement).

Note: If you have been using Eclipse with ADT, be aware that [Android Studio](#) is now the official IDE for Android, so you should migrate to Android Studio to receive all the latest IDE updates. For help moving projects, see [Migrating to Android Studio](#).

ADT n'étant plus mise à jour et Google recommandant son produit pour obtenir les dernières mises à jour, Android Studio a donc été choisi pour développer le projet.

Pour la partie base de données, SQL Lite est la solution de base implémentée pour Android. Cependant les recherches ont mises en avant un grand nombre de développeurs ayant des problèmes et principalement des problèmes liés à la vitesse d'exécution des requêtes ainsi qu'à la difficulté d'implémentation des bases de données avec SQL Lite. Comme solutions, beaucoup de réponses parlaient et mettaient en avant Realm qui une librairie permettant la création de base de données sur mobile. Pourquoi choisir Realm plutôt que SQL Lite ou un ORM¹ comme greenDAO ? Premièrement, car Realm est facile à implémenter, il utilise son propre mécanisme de persistance² et est capable de sauvegarde n'importe quel objet implémenté dans le code. Deuxièmement, car Realm est multiplateformes. Un fichier *.realm contenant une base de données peut très facilement être importée dans un autre projet que ce soit un projet Android ou IOS. Et enfin, car il est, d'après le benchmark trouvé, plus rapide qu'un ORM ou qu'une base de données SQL Lite. Realm est capable d'insérer 2000 enregistrements à la seconde contre 900 pour SQL Lite et 500 pour OrmLite et exécuter 51 requêtes secondes contre 7 pour les autres plateformes même s'il faut bien sûr prendre du recul sur ces résultats, car ils ont été effectués par les créateurs de Realm et non par un tiers.

Benchmark tiré de : <http://realm.io/news/realm-for-android/#realm-for-android>



3.1 Requêtes avec Realm (51 par sec)

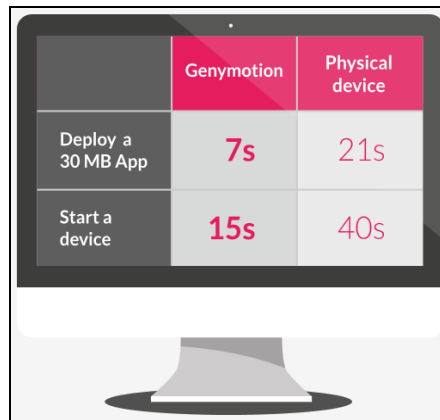
3.2 Insertions avec Realm (2k par sec)

Pour la compilation du projet, un samsung S3 version 4.3 a été utilisé ainsi que GenyMotion pour émuler des autres terminaux. L'émulateur d'android permet l'installation de nouveau terminal, mais la mise en place est longue et complexe pour un résultat et une rapidité très moyenne. GenyMotion est une solution rapide et facile à mettre en place puisqu'un plugin est directement téléchargeable depuis Android Studio. Il s'intègre dans la barre des tâches et propose une liste de plus de 80 devices dans 7 versions d'android différentes. Pour les installer rien de plus simple car se sont de simples Machines Virtuelles gérées grâce à VirtualBox. Le téléchargement de la machine virtuelle suffit à avoir un émulateur prêt à être lancé. Un gain de temps

¹ Un ORM (Object-relational mapping) ou en français mapping objet-relationnel est une technique de programmation qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle.

² La persistance en programmation informatique se réfère au mécanisme responsable de la sauvegarde et de la restauration des données.

énorme pour développer et tester son application sur un grand nombre de devices et ainsi augmenter la compatibilité, sans mettre en avant la possibilité de prendre des screenshots, de simuler des positions GPS, d'installer des applications par Drag and Drop ou encore de simuler les capteurs comme l'accéléromètre.



	Genymotion	Physical device
Deploy a 30 MB App	7s	21s
Start a device	15s	40s

3.3 Comparaison GenyMotion

Concernant le reste des logiciels utilisés, ils sont les suivants : Microsoft Office 2016 en preview pour la gestion de la partie administrative et rapport du projet, Gimp et Inkscape pour tout ce qui est la partie design et retouche (principalement pour les interfaces), Chrome pour la navigation et eduncanet, qui est la plateforme de l'ETML pour la correspondance. Tous ces outils sont des outils gratuits et ils ont été choisis pour cela.

3.2 Document d'analyse et conception

- Ce paragraphe décrit le fonctionnement de manière détaillée.
 - Autant que possible de manière graphique
 - Tous les cas particuliers devraient y être spécifiés...
- Il s'agit d'y présenter les fonctionnalités à développer:
 - Découpage en étapes, en modules, en fonctionnalités, etc.
 - Formulaire, interfaces graphiques, pages web, etc.
 - Schémas de navigation, schémas événementiels, structogramme, pseudo-code, etc.
- Si le projet inclut une base de données:
 - Dictionnaire des données
 - Modèle conceptuel des données, modèles logique des données.

3.3 Conception des tests

- Ce paragraphe permet de spécifier la stratégie de test:
 - Qui, quand, avec quelles données, dans quel ordre, etc.

3.4 Planning détaillé

- A ce stade, après l'analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, ...) peut être finalisé.

4 Réalisation

4.1 Dossier de réalisation

- Ce paragraphe permet de reproduire ou reprendre le projet par un tiers.

Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement:

- Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)
- Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)
- Code source des éléments logiciels développés.
- Modèle physique d'une base de données.
- Arborescences des documents produits.
- Il faut décrire le parcours de réalisation et justifier les choix.

4.2 Modifications

- Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.
 - Date, raison, description, etc.

5 Tests

5.1 Dossier des tests

- On dresse le bilan des tests effectués (qui, quand, avec quelles données...) sous forme de procédure (tableau).
- Si des tests prévus dans la stratégie n'ont pas pu être effectués:
 - raison, décisions, etc.
- Liste des bugs répertoriés avec la date de découverte et leur état:
 - Corrigé, date de correction, corrigé par, etc.

6 Conclusion

6.1 Bilan des fonctionnalités demandées

- Il s'agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.
- Si ce n'est pas le cas, mesuré en « % » ou en « temps supplémentaire » le travail qu'il reste à accomplir pour terminer le tout.

6.2 Bilan de la planification

- Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet.

6.3 Bilan personnel

- Si c'était à refaire:
 - Qu'est-ce qu'il faudrait garder ? Les plus et les moins ?
 - Qu'est-ce qu'il faudrait gérer, réaliser ou traiter différemment ?
- Qu'est-ce que ce projet m'a appris ?
- Suite à donner, améliorations souhaitables, ...
- Remerciements, signature, etc.

7 Divers

7.1 Journal de travail de chaque participant

- Date, activité (décrit afin de reproduire le cheminement du projet), durée.

7.2 Bibliographie

- Références des livres utilisés durant le projet.

7.3 Webographie

- Références des sites Internet consultés durant le projet.

8 Annexes

- Listing du code source (partiel ou, plus rarement complet)
- Guide(s) d'utilisation et/ou guide de l'administrateur
- Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).
- Extraits de catalogue, documentation de fabricant, etc.
- Photocopies diverses, etc.