

Documentation interne

Projet de Programmation

L3L1



Illustration 1: Image de présentation

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

Les informations d'identification du document :

Référence du document :
Version du document : 1
Date du document : 20/04/2021
Auteurs : Youstina Abdel Massih Thierno Bah Pathé Mbaye Ndéye Mehdi Hamiche

Les éléments de vérification du document :

Validé par :
Validé le :
Soumis le :
Type de diffusion :
Confidentialité :

Les éléments d'authentification :

Maître d'ouvrage :	Chef de projet :
Date / Signature :	Date / Signature :

Youstina Abdel Massih
Thierno Bah
Pathe Mbaye Ndéye
Mehdi Hamiche

Groupe L3L1

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

Sommaire

<u>1. Introduction</u>	6
1.1. Objectifs et méthodes	6
1.2. Documents de référence	7
<u>2. Guide de lecture</u>	8
2.1. Maîtrise d'œuvre	8
<i>2.1.1. Responsable</i>	8
<i>2.1.2. Personnel administratif</i>	8
<i>2.1.3. Personnel technique</i>	8
2.2. Maîtrise d'ouvrage	9
<i>2.2.1. Responsable</i>	9
<i>2.2.2. Personnel administratif</i>	9
<i>2.2.3. Personnel technique</i>	9
<u>3. Concepts de base</u>	10
<u>4. Description des classes pour chaque fonctionnalités</u>	11
4.1. Menu principal	11

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

<i>4.1.1. Classes associés</i>	<i>11</i>
<i>4.1.2. Interfaces graphiques associés</i>	<i>11</i>
4.2. Dessin du tracé en temps réel	<i>11</i>
<i>4.2.1. Classes associés</i>	<i>12</i>
<i>4.2.2. Interfaces graphiques associés</i>	<i>12</i>
4.3.Télécharger un tracé	<i>12</i>
<i>4.3.1. Classes associés</i>	<i>13</i>
<i>4.2.2. Interfaces graphiques associés</i>	<i>13</i>
4.4. Enregistrer un tracé	<i>13</i>
<i>4.4.1. Classes associés</i>	<i>14</i>
<i>4.4.2. Interfaces graphiques associés</i>	<i>14</i>
4.5. Menu Historique	<i>14</i>
<i>4.5.1. Classes associés</i>	<i>14</i>
<i>4.5.2. Interfaces graphiques associés</i>	<i>15</i>
4.6. Favoris	<i>16</i>
<i>4.6.1. Classes associés</i>	<i>16</i>

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

<i>4.6.2. Interfaces graphiques associés</i>	<i>16</i>
<u>6. Annexes</u>	<i>18</i>
<u>7. Glossaire</u>	<i>19</i>
<u>8. Références</u>	<i>20</i>
<u>9. Index</u>	<i>21</i>

1. Introduction

Cette documentation facilite la maintenance de l'application par un tiers. Celle-ci peut prendre plusieurs formes en fonction du langage utilisé pour l'implantation dont Javadoc pour Java.

Toutefois, un document généré automatiquement à l'aide de ces outils est insuffisant.

Contenu :

- *Concept de base*
 - permet la compréhension du document
- *Description des modules (classes)*
 1. Rappel des objectifs du module
 2. Relations d'utilisation avec d'autres modules
 3. Lister les modules utilisés par ce module et ceux utilisant ce module
 4. Procédures externes
 5. Variables externes

1.1. Objectifs et méthodes

1. *Implémentation de l'ensemble des interfaces graphiques*
2. *Détection des capteurs du téléphone*
3. *Implémentation du tracé sensible*
4. *Téléchargement d'une photo d'un tracé dans la galerie du téléphone*

5. *Implémentation d'un formulaire pour enregistrer les informations d'un tracé*

1.2. Documents de référence

Les documents du projet servant à l'élaboration du présent document :

- Le cahier des charges,
- Le cahier de recettes,
- La conception générale
 - ◆ Conception détaillée
 - ◆ Manuel d'installation
 - ◆ Manuel d'utilisation
 - ◆ Plan de tests

2. Guide de lecture

2.1. Maîtrise d'œuvre

La maîtrise d'œuvre présente l'équipe du développement chargé du bon suivi de la documentation interne et des besoins dont le maître d'ouvrage fait commande.

Elle représente l'équipe du développement :

- Youstina Abdel Massih
- Thierno Bah
- Pathé Mbaye Ndéye
- Mehdi Hamiche

Cette équipe veillera au bon suivi de la documentation interne coordonnées avec la conception générale représentant les besoins de l'enseignante encadrante.

2.1.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'œuvre de lire le document dans sa totalité afin de prendre conscience de l'ensemble des éléments.

2.1.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire le concept de base et la description des modules.

2.1.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie sur le concept de base, ainsi que la description des modules.

2.2. Maîtrise d'ouvrage

La maîtrise d'ouvrage représente dans notre cas le client du projet, c'est-à-dire les personnes dont les besoins permettent la conception du projet.

La maîtrise d'ouvrage est assistée par l'équipe de la maîtrise d'œuvre et donc ce rôle sera assuré par l'enseignante encadrante Flavie Tonon.

2.2.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'ouvrage de lire le document dans toute sa totalité afin de prendre conscience de l'ensemble des documents.

2.2.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire le concept de base et la description des modules.

2.2.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie sur le concept de base, ainsi que la description des modules.

3. Concepts de base

Pour bien comprendre ce document, lire tous les documents liés à la conception générale pour avoir les notions sur les différents modules abordés.

4. Description des classes pour chaque fonctionnalités

4.1. Menu principal

Le menu principal est composé des boutons « *Démarrer le tracé* » et « *Historique* » qui permettent respectivement de démarrer un tracé et d'accéder à l'historique de l'application.

4.1.1. Classes associés

Pour le menu principal, voici la classe associée et leur fonction :

MainActivity.java - Classe principale de notre projet. Elle donne des actions aux deux boutons du menu principal. En effet, elle permet par l'intermédiaire du bouton « *Démarrer le tracé* », de commencer le dessin d'un tracé sensible. De plus, elle permet au bouton « *Historique* » de nous faire accéder à l'historique de l'application

4.1.2. Interfaces graphiques associés

Pour le menu principal, il y a une seule interface graphique associée :

activity_main.xml – Cette interface nous a permis de définir l'emplacement des boutons et le design du menu principal

4.2. Dessin du tracé en temps réel

Le dessin du tracé en temps réel est possible grâce au bouton « *Démarrer le tracé* » sur le menu principal de l'application.

4.2.1. Classes associées

Pour le dessin du tracé en temps réel, voici les classe associées et leurs fonctions :

MainActivity.java - Classe principale de notre projet. Elle donne des actions aux deux boutons du menu principal. En effet, elle permet par l'intermédiaire du bouton « Démarrer le tracé », de commencer le dessin d'un tracé sensible. De plus, elle permet au bouton « Historique » de nous faire accéder à l'historique de l'application

Point.java - Cette classe a permis de définir chaque point du tracé

Trace.java - Cette classe permet de dessiner le tracé en fonction des mouvements du téléphone, et à l'aide de la classe **Point**

TraceActivity.java - Cette classe permet le dessin du tracé, le téléchargement d'une image du tracé dans la galerie du téléphone et l'enregistrement du tracé dans la base de données de l'application. En effet, cette classe gère toutes les activités qui peuvent être effectuées sur la page du tracé

4.2.2. Interfaces graphiques associés

Pour le dessin du tracé en temps réel, il y a une seule interface graphique associée :

activity_trace.xml - Cette interface nous a permis de définir l'emplacement des boutons et de la zone où le tracé se dessine

4.3. Télécharger un tracé

Le téléchargement d'un tracé est possible après avoir appuyé sur le bouton « Démarrer le tracé ». L'utilisateur est redirigé vers la page du tracé dans lequel il y

aura le dessin du tracé sensible en temps réel, et les boutons « *Télécharger* » et « *Enregistrer* ». Lorsque l'utilisateur appuiera sur le bouton « *Télécharger* », il aura un message de confirmation. Ainsi il pourra retrouver une image du tracé dans la galerie de son téléphone.

4.3.1. Classes associés

Pour le téléchargement d'un tracé voici la classe associée et leur fonction :

TraceActivity.java - Cette classe permet le dessin du tracé, le téléchargement d'une image du tracé dans la galerie du téléphone et l'enregistrement du tracé dans la base de données de l'application. En effet, cette classe gère toutes les activités qui peuvent être effectuées sur la page du tracé.

4.2.2. Interfaces graphiques associés

Pour le téléchargement d'un tracé, il y a une seule interface graphique associée :

activity_trace.xml - Cette interface nous a permis de définir l'emplacement des boutons et de la zone où le tracé se dessine

4.4. Enregistrer un tracé

L'enregistrement d'un tracé est possible après avoir appuyé sur le bouton « *Démarrer le tracé* ». L'utilisateur est redirigé vers la page du tracé dans lequel il y aura le dessin du tracé sensible en temps réel, et les boutons « *Télécharger* » et « *Enregistrer* ». Lorsque l'utilisateur appuiera sur le bouton « *Enregistrer* », il sera redirigé vers la page du formulaire où il devra entrer les informations associées à son tracé. Enfin, l'utilisateur devra appuyer sur « *Valider* » pour finaliser l'enregistrement dans l'historique de l'application.

4.4.1. Classes associés

Pour l'enregistrement d'un tracé, voici les classes associées et leurs fonctions :

TraceActivity.java - Cette classe permet le dessin du tracé, le téléchargement d'une image du tracé dans la galerie du téléphone et l'enregistrement du tracé dans la base de données de l'application. En effet, cette classe gère toutes les activités qui peuvent être effectuées sur la page du tracé.

FormulaireActivity.java - Cette classe permet de gérer les entrées de la date du tracé et du mode de transport utilisé par l'utilisateur.

4.4.2. Interfaces graphiques associés

Pour l'enregistrement d'un tracé, il y a deux interfaces graphiques associées :

activity_trace.xml - Cette interface nous a permis de définir l'emplacement des boutons et de la zone où le tracé se dessine

activity_formulaire.xml - Cette interface nous a permis de définir l'emplacement des champs à remplir et du bouton « Valider » dans la page de formulaire

4.5. Menu Historique

L'accès à l'historique est possible grâce au bouton « Historique » sur le menu principal de l'application.

4.5.1. Classes associés

Pour accéder à l'historique, voici les classes associées et leurs fonctions :

MainActivity.java - Classe principale de notre projet. Elle donne des actions aux deux boutons du menu principal. En effet, elle permet par l'intermédiaire du bouton «

Démarrer le tracé », de commencer le dessin d'un tracé sensible. De plus, elle permet au bouton « *Historique* » de nous faire accéder à l'historique de l'application

HistoriqueActivity.java - Cette classe permet d'afficher l'image d'un tracé dans l'historique et dans les favoris en grand, de supprimer un tracé de l'historique, et de rajouter un tracé dans les favoris

HistoriqueViewAdapter.java - Cette classe permet de gérer l'affichage de la liste des tracés dans l'historique et dans les favoris. Cette classe permet aussi de supprimer un tracé des favoris

AfficherTrajetHistoriqueActivity.java - Cette classe permet d'afficher une photo d'un tracé dans l'historique et dans les favoris en grand, puis de fermer l'affichage de la photo (pour revenir à la liste des tracés dans l'historique et dans les favoris)

KourteDbHelper.java - Cette classe correspond à la base de données SQLite de l'application qui permet de stocker toutes les informations liées à un tracé afin de pouvoir le retrouver dans l'historique et dans les favoris de l'application

4.5.2. Interfaces graphiques associés

Pour l'historique, voici les interfaces graphiques associées :

activity_main.xml - Cette interface nous a permis de définir l'emplacement des boutons et le design du menu principal

row_historique.xml - Cette interface a permis de définir l'affichage de manière indépendante chaque élément dans la liste de l'historique

activity_historique.xml – Cette interface a permis de définir l'affichage de l'ensemble des éléments de l'historique à l'aide de row_historique.xml

4.6. Favoris

4.6.1. Classes associées

Pour les favoris, voici les classes associées et leurs fonctions :

MainActivity.java - Classe principale de notre projet. Elle donne des actions aux deux boutons du menu principal. En effet, elle permet par l'intermédiaire du bouton « Démarrer le tracé », de commencer le dessin d'un tracé sensible. De plus, elle permet au bouton « Historique » de nous faire accéder à l'historique de l'application

HistoriqueActivity.java - Cette classe permet d'afficher l'image d'un tracé dans l'historique et dans les favoris en grand, de supprimer un tracé de l'historique, et de rajouter un tracé dans les favoris

HistoriqueViewAdapter.java - Cette classe permet de gérer l'affichage de la liste des tracés dans l'historique et dans les favoris. Cette classe permet aussi de supprimer un tracé des favoris

AfficherTrajetHistoriqueActivity.java - Cette classe permet d'afficher une photo d'un tracé dans l'historique et dans les favoris en grand, puis de fermer l'affichage de la photo (pour revenir à la liste des tracés dans l'historique et dans les favoris)

KourteDbHelper.java - Cette classe correspond à la base de données SQLite de l'application qui permet de stocker toutes les informations liées à un tracé afin de pouvoir le retrouver dans l'historique et dans les favoris de l'application

4.6.2. Interfaces graphiques associés

Pour les favoris, voici les interfaces graphiques associées :

activity_main.xml - Cette interface nous a permis de définir l'emplacement des boutons et le design du menu principal

row_historique.xml - Cette interface a permis de définir l'affichage de manière indépendante chaque élément dans la liste de l'historique

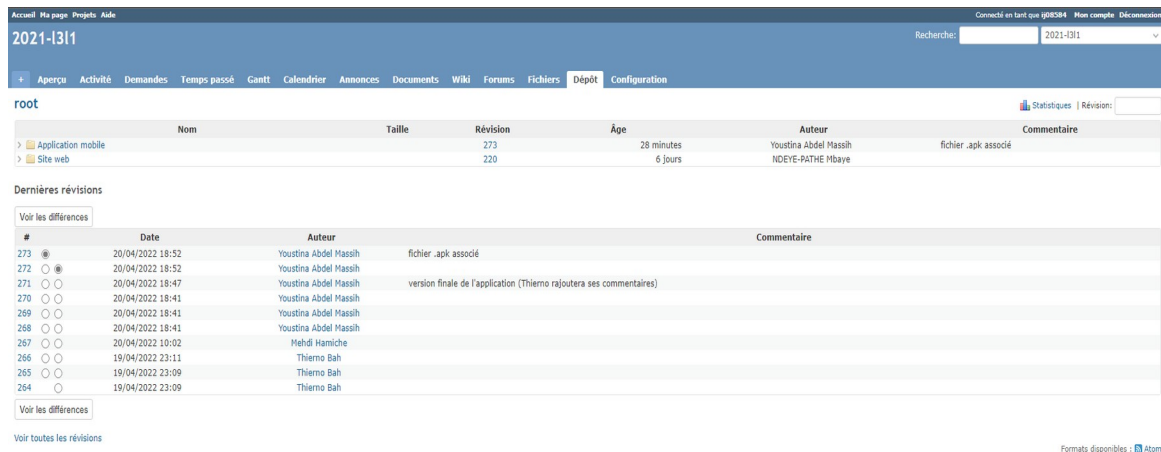
activity_historique.xml – Cette interface a permis de définir l'affichage de l'ensemble des éléments de l'historique à l'aide de row_historique.xml

menu.xml - Cette interface a permis d'afficher les éléments qui ont été placés dans les favoris et de revenir à l'historique

6. Annexes

➤ Forge

➤ <https://forge.ens.math-info.univ-paris5.fr/projects/2021-l3l1/repository>



2021-l3l1

Recherche: 2021-l3l1

Accueil Ma page Projets Aide

2021-l3l1

Rechercher: 2021-l3l1

Statistiques Révisions

Nom	Taille	Révision	Âge	Auteur	Commentaire
Application mobile		273	28 minutes	Youstina Abdel Massih	fichier .apk associé
Site web		220	6 jours	NDEYE-PATHE Mbaye	

Dernières révisions

Voir les différences

#	Date	Auteur	Commentaire
273	20/04/2022 18:52	Youstina Abdel Massih	fichier .apk associé
272	20/04/2022 18:52	Youstina Abdel Massih	
271	20/04/2022 18:47	Youstina Abdel Massih	version finale de l'application (Thierno rajouter ses commentaires)
270	20/04/2022 18:41	Youstina Abdel Massih	
269	20/04/2022 18:41	Youstina Abdel Massih	
268	20/04/2022 18:41	Youstina Abdel Massih	
267	20/04/2022 18:02	Mehdi Hamiche	
266	19/04/2022 23:11	Thierno Bah	
265	19/04/2022 23:09	Thierno Bah	
264	19/04/2022 23:09	Thierno Bah	

Voir toutes les révisions

Formats disponibles : Atom

Illustration 2: Forge - partie où se trouve le projet

7. Glossaire

- ★ **DOCUMENTATION INTERNE** : Consiste à rédiger et à tenir à jour des processus et procédures clairement détaillés, auxquels les membres de votre équipe interne peuvent se référer
- ★ **APPLICATION MOBILE** : Logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un smartphone, un baladeur numérique, une tablette tactile...
- ★ **APPLICATION** : Programme ou ensemble de programmes destiné à aider l'utilisateur d'un ordinateur pour le traitement d'une tâche précise
- ★ **CODE SOURCE** : Code écrit dans un langage de programmation et qui peut être converti pour constituer un programme exécutable
- ★ **.APK** : de l'acronyme Android PacKage, extension de fichier Équivalent du ".exe" sur Windows
- ★ **ANDROID** : Logiciel géant pour téléphones portables, tablettes et bien d'autres appareils qui peuvent être intelligents, comme par exemple certaines voitures. C'est le système d'exploitation le plus utilisé au monde sur mobile. Il permet la communication entre un utilisateur et son appareil

8. Références

<https://developer.android.com/docs>

<https://developer.android.com/guide/topics/ui/declaring-layout>

<https://forge.ens.math-info.univ-paris5.fr/projects/2021-l3l1/repository>

9. Index

Index des figures

<i>Illustration 1: Image de présentation</i>	<i>1</i>
<i>Illustration 2: Forge - partie où se trouve le projet</i>	<i>18</i>

Index lexical

application	6, 11 sqq.
APPLICATION	19
base de données	12 sqq.
boutons	11 sqq.
capteurs	6
classe	6, 11 sqq.
Classe	11 sqq., 16
conception générale	7 sq., 10
Démarrer le tracé	11 sqq., 15 sq.
enregistrement	12 sqq.
enregistrer	7
Enregistrer	13

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

formulaire	7
historique	11 sqq.
Historique	11 sq., 14 sqq.
image	12 sqq.
Implémentation	6 sq.
informations	7
interface	6, 11 sqq.
Interface	11 sqq.
java	11 sqq.
Java	6
Javadoc	6
l'application	6
menu principal	11 sq., 14 sqq.
Menu principal	11
méthodes	6
mouvements	12
point	12

Projet L3L - Kourte : une écriture des forces physiques, le temps d'un trajet

Point	12
sensible	6, 11 sqq., 15 sq.
Télécharger	12 sq.
téléphone	6
tracé	6 sq., 11 sqq.
Valider	13 sq.