

Conception détaillée

Projet de Programmation

L3L1

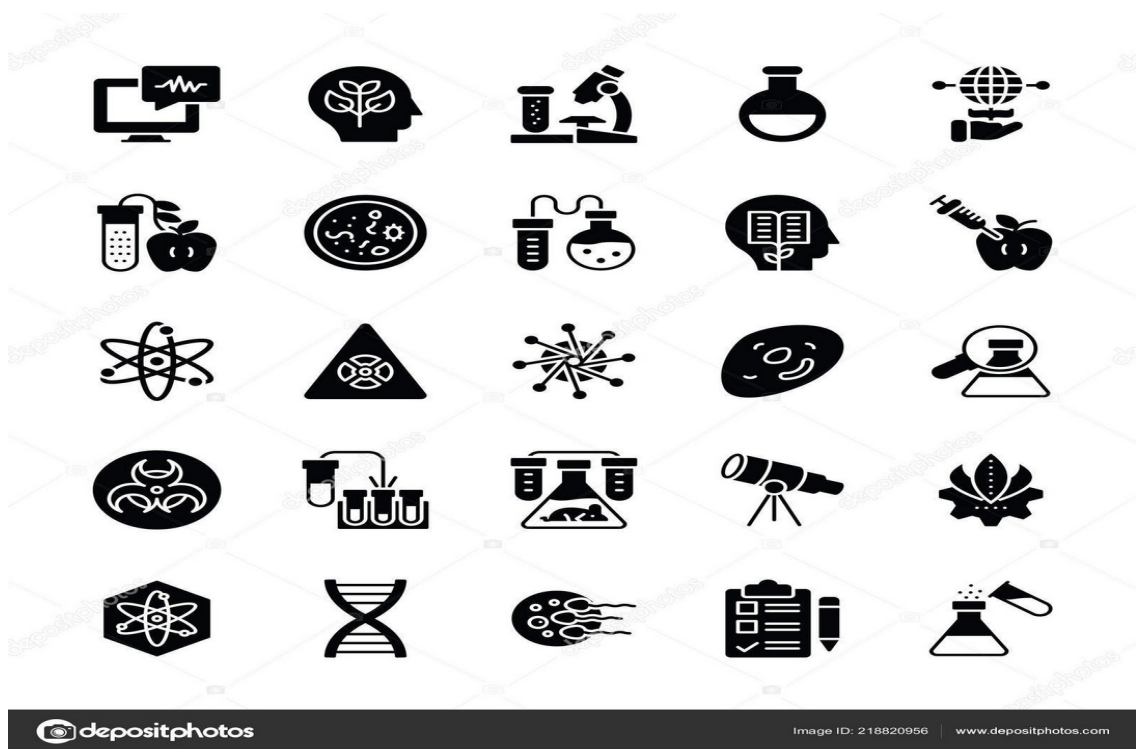


Illustration 1: Image de présentation

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

Les informations d'identification du document :

Les éléments de vérification du document :

| |
|--|
| Référence du document : |
| Version du document : 3 |
| Date du document : 20/04/2022 |
| Auteurs : Youstina Abdel Massih Thierno Bah Pathé Mbaye Ndéye Mehdi Hamiche |

| |
|----------------------------|
| 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
Pathé Mbaye Ndéye
Mehdi Hamiche

Groupe L3L1

Sommaire

| | |
|--|-----------|
| <u>1. Introduction</u> | 5 |
| <u>2. Guide de lecture</u> | 6 |
| 2.1. Maîtrise d'œuvre | 6 |
| <i>2.1.1. Responsable</i> | 6 |
| <i>2.1.2. Personnel administratif</i> | 6 |
| <i>2.1.3. Personnel technique</i> | 6 |
| 2.2. Maîtrise d'ouvrage | 7 |
| <i>2.2.1. Responsable</i> | 7 |
| <i>2.2.2. Personnel administratif</i> | 7 |
| <i>2.2.3. Personnel technique</i> | 7 |
| <u>3. Conception du projet</u> | 8 |
| 3.1. Architecture du projet | 8 |
| 3.2. Diagramme de cas d'utilisation | 10 |
| 3.3. Diagramme de classes | 12 |
| 3.4. Cycle de vie | 14 |

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

| | |
|--|-----------|
| <u>4. Description du module</u> | 15 |
| 4.1. Rappel des objectifs du composant | 15 |
| 4.2. Composantes de l'application et du site web | 15 |
| <i>4.2.1. Définir les différentes interfaces et leurs visibilité</i> | 15 |
| <i>4.2.2. Décomposition en tâches</i> | 18 |
| <i>4.2.3. Méthodes et outils</i> | 19 |
| <i>4.2.4. Standard et outils</i> | 19 |
| <i>4.2.5. Références</i> | 19 |
| <u>4.3. Les classes et librairies pour l'application</u> | 20 |
| <u>5. Annexes</u> | 22 |
| <u>6. Glossaire</u> | 23 |
| <u>7. Références</u> | 24 |
| <u>8. Index</u> | 25 |

1. Introduction

La conception détaillée affine la conception générale en présentant toutes les fonctions, méthodes, classes, paquetages, et librairies qui seront nécessaires au bon développement de l'application mobile et du site web.

Ce document vise à faciliter l'implémentation de l'application et du site web par les développeurs et vise à garantir que le fonctionnement de l'application et du site web correspondra bien aux besoins de l'utilisateur final, ce qui permettrait au développeur de pouvoir réaliser facilement une traduction de ce document en langage de programmation.

Ce document est aussi organisé en fonction de l'architecture de l'application et du site web, en répétant autant de fois que nécessaire les rubriques suivantes. Pour chaque composant logiciel issu de la conception générale.

2. Guide de lecture

2.1. Maîtrise d'œuvre

La maîtrise d'œuvre présente l'équipe du développement chargé du bon suivi de la conception détaillée 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 conception détaillée coordonnées avec le 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 la présentation du produit ainsi que la décomposition des tâches.

2.1.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie de la description du module, avec la liste des composants utilisés par ce composant et ceux

utilisant ce composant, ainsi que la liste des classes / structures de données / méthodes / fonctions du composant.

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 la présentation du produit ainsi que la décomposition des tâches.

2.2.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie de la description du module, avec la liste des composants utilisés par ce composant et ceux utilisant ce composant, ainsi que la liste des classes / structures de données / méthodes / fonctions du composant.

3. Conception du projet

Afin de réaliser le projet nous utiliserons les outils et langages suivants :

- Android Studio est un environnement de développement qui nous permettra de développer l'application mobile
- JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permettra de représenter l'information de façon structurée lors de la communication entre le SGBD et le serveur
- MySQL permet la sauvegarde des informations
- PHP permettra la demande de ressource au serveur par le navigateur web
- Le logiciel libre Apache HTTP Server (Apache) est un serveur HTTP qui va servir d'intermédiaire entre le SGBD et l'application en exécutant certaines tâches comme l'envoi d'information en HTTP, la demande de requête en SQL pour la base de données mais aussi la réception des informations de ces derniers
- Java est le langage de programmation sur lequel la majorité du code de l'application reposera

3.1. Architecture du projet

Pour l'architecture du projet, on a le diagramme de déploiement qui nous permet de représenter l'utilisation de l'infrastructure physique par le système et la manière dont les **composants** du système sont répartis ainsi que leurs relations entre eux. Les

éléments utilisés par un **diagramme de déploiement** sont principalement les **nœuds**, les **composants**, les **associations** et les **artefacts**.

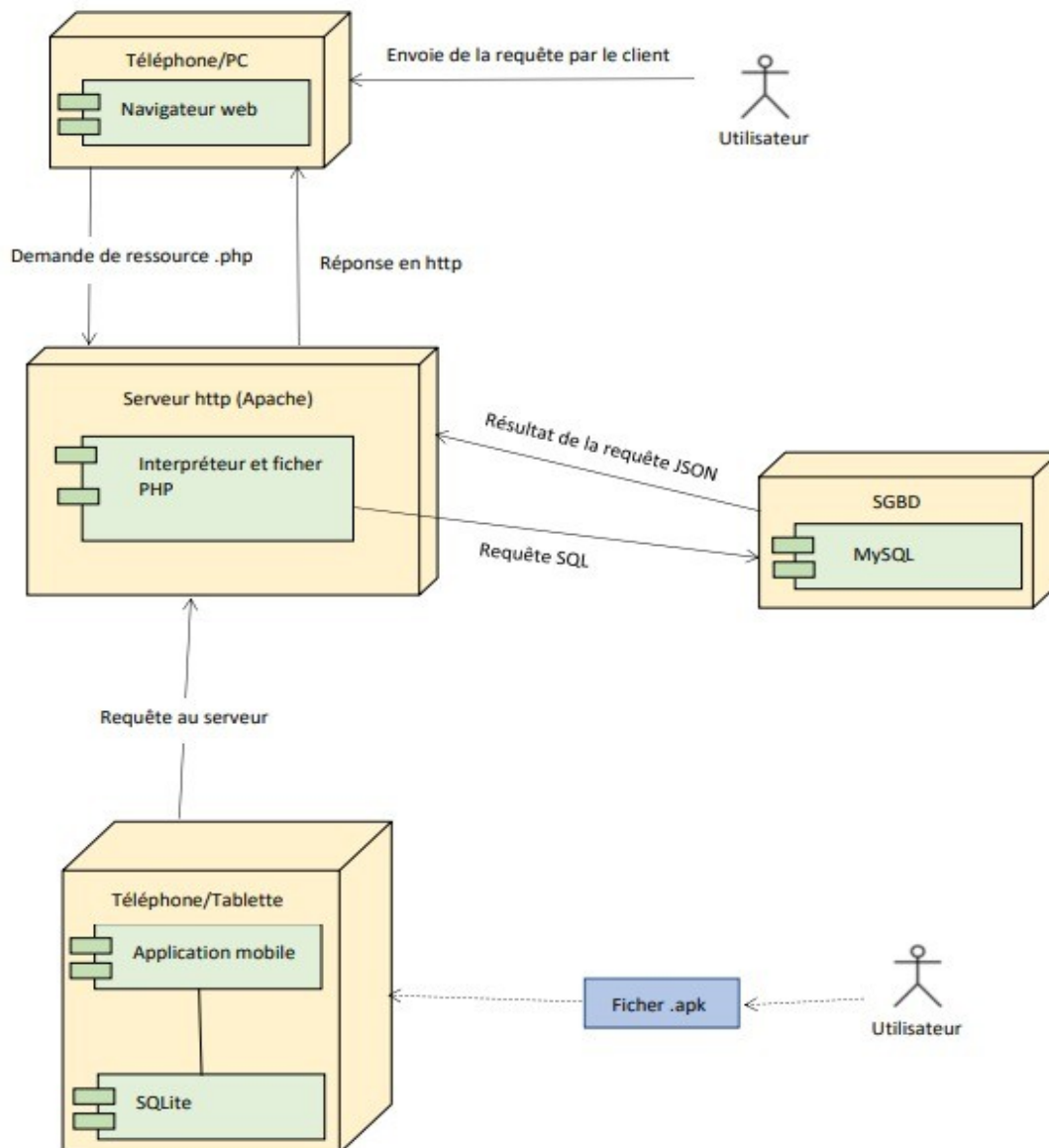


Illustration 2: Diagramme de déploiement

3.2. Diagramme de cas d'utilisation

Dans notre projet, les diagrammes de cas d'utilisation permettent de représenter le comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

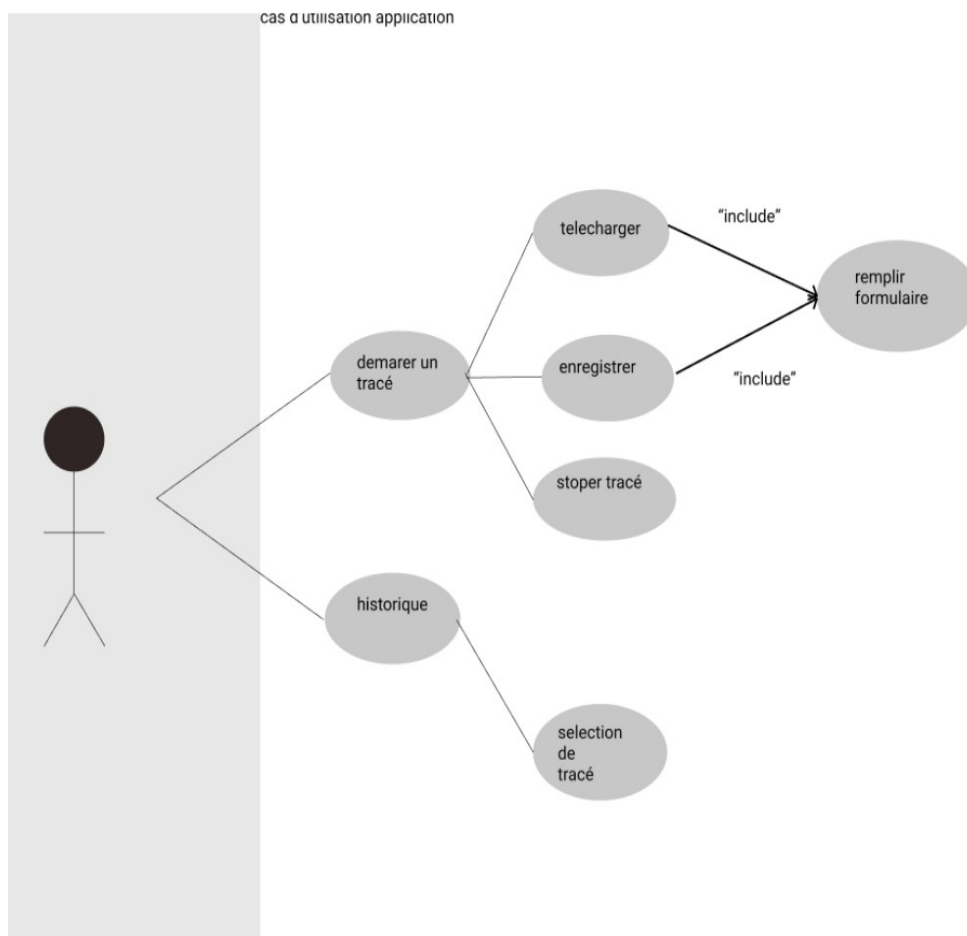


Illustration 3: Diagramme de cas d'utilisation (application)

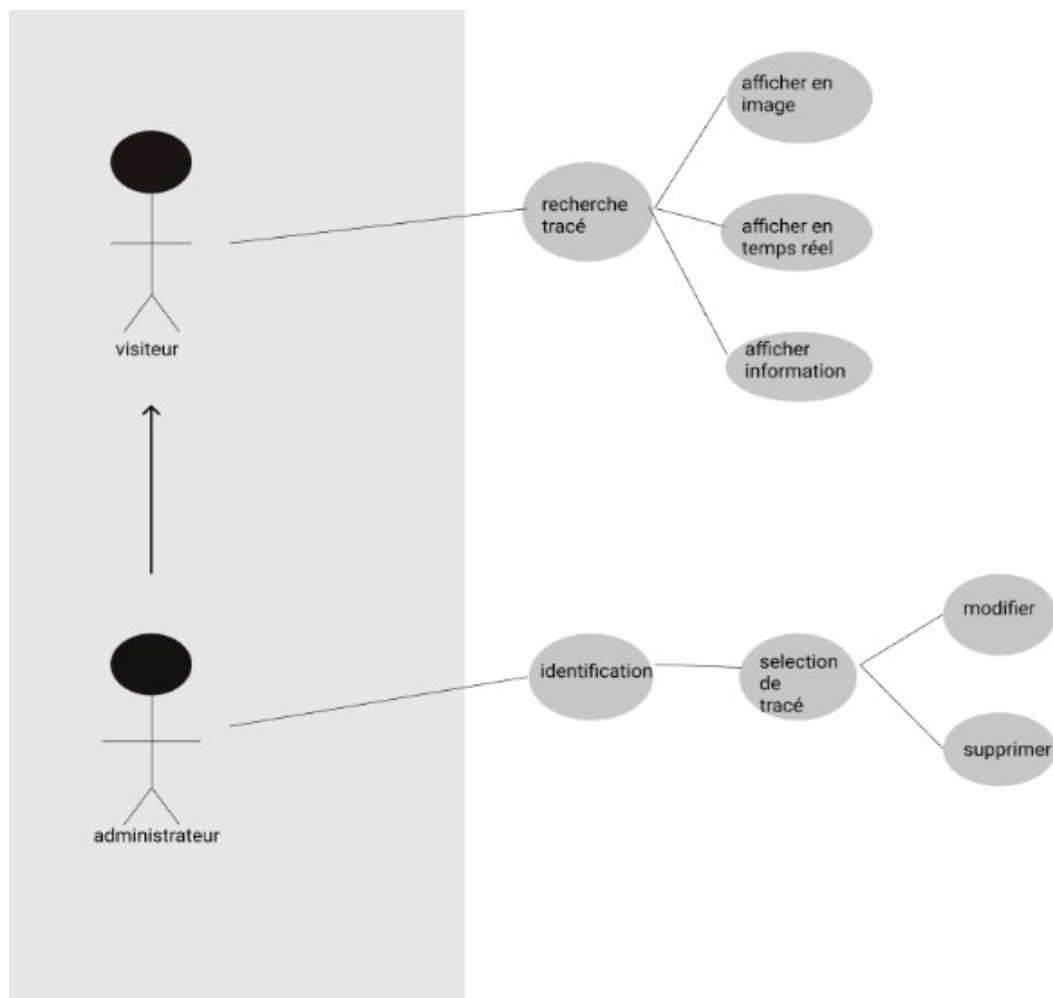


Illustration 4: Diagramme de cas d'utilisation (site web)

3.3. Diagramme de classes

Dans notre projet, nous utilisons une programmation orientée objet. On utilise alors des classes que nous modélisons à partir du diagramme de classes. Celui-ci se présente sous la forme d'un schéma UML pour présenter les classes, leurs fonctions, attributs et leur lien avec les autres classes.

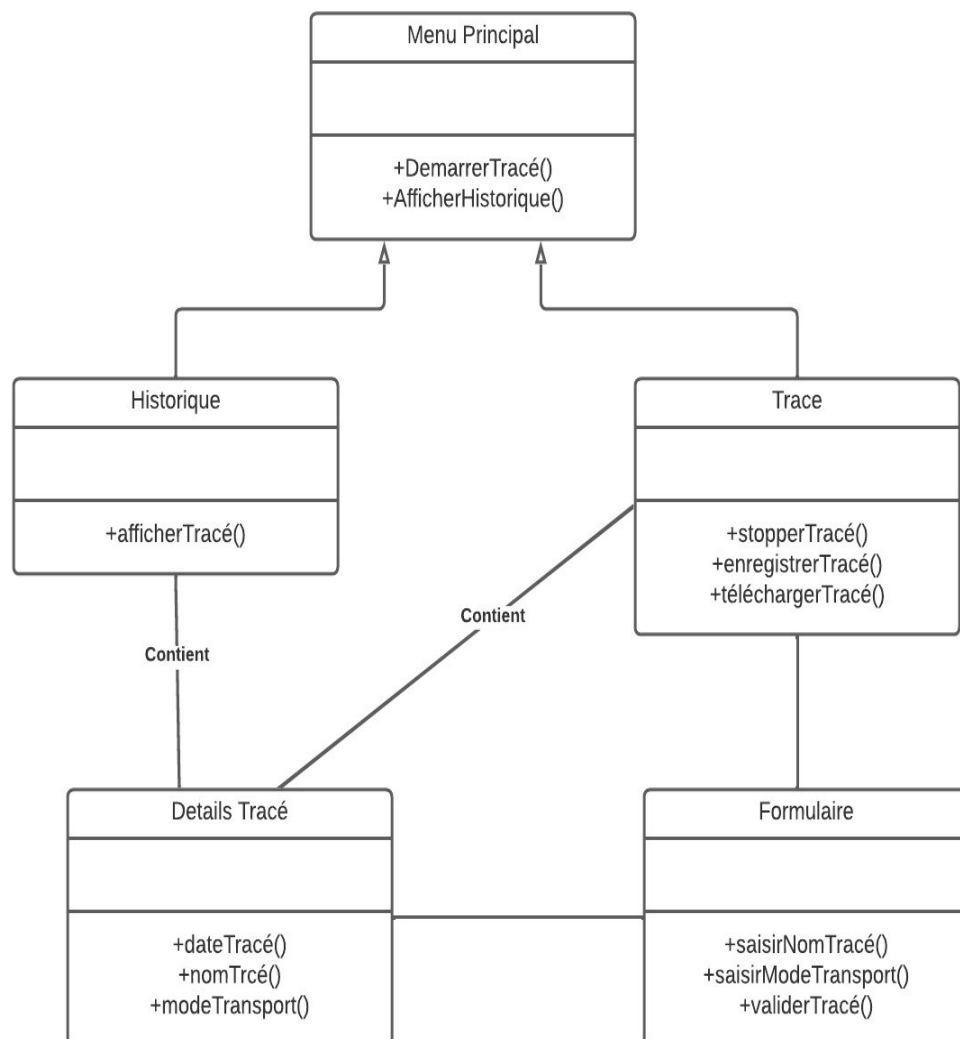


Illustration 5: Diagramme de classes (Application)

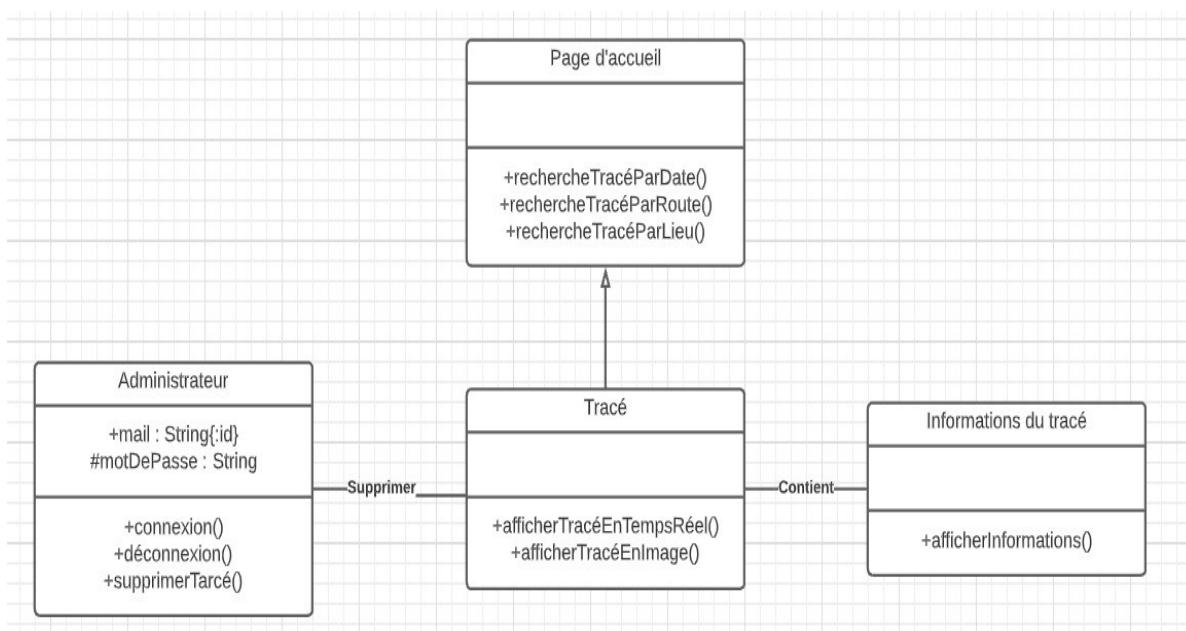


Illustration 6: Diagramme de classes (Site web)

3.4. Cycle de vie

Dans notre projet, nous allons utiliser le modèle de cycle en V car avec ce modèle, notre projet serait géré de façon organisée en appliquant un planning de conceptions et de validations pour éviter au maximum les retours en arrière. Chaque étape du projet doit être validée avant de débiter l'étape suivante.

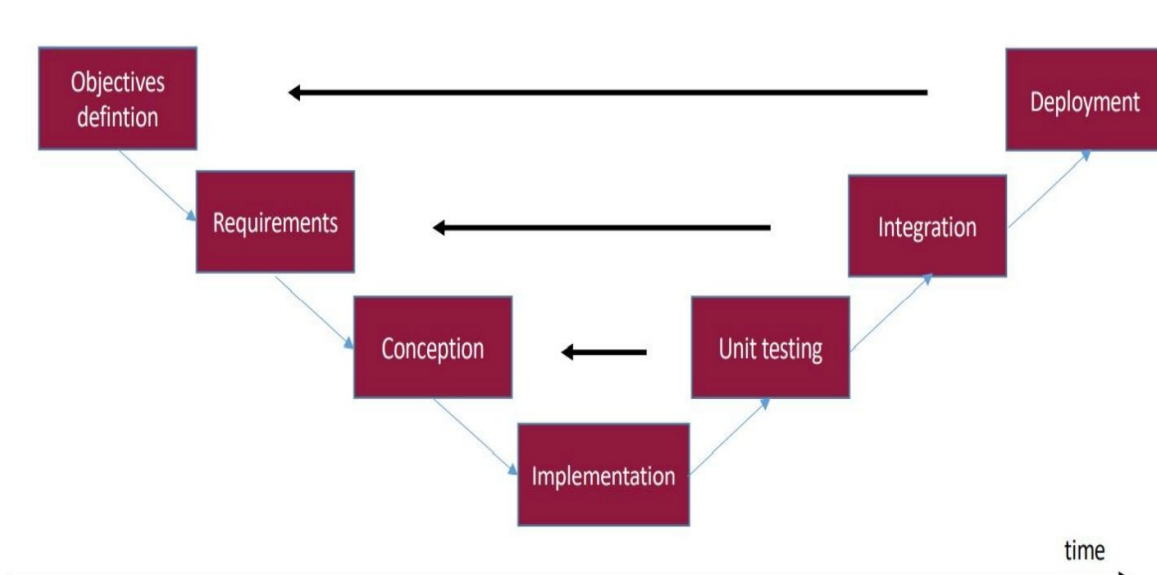


Illustration 7: Modèle cycle de vie en V

4. Description du module

4.1. Rappel des objectifs du composant

Le produit que nous réalisons sous forme d'application mobile (fichier d'extension .apk) et de site web, sera livré en état de marche avec son code source. La documentation du produit contiendra les manuels d'utilisation et d'installation, le cahier des charges, le cahier de recettes, et d'autres documents nécessaires au projet.

4.2. Composantes de l'application et du site web

Pour pouvoir utiliser l'application mobile et le site web, l'élément principal qui leur est lié est d'avoir un ordinateur dont la version des mises à jour soit assez récente, cela permettrait de pouvoir utiliser totalement toutes les fonctionnalités et d'en avoir les meilleures performances ainsi qu'un téléphone Android (Version 7.0 pour utiliser au mieux les capteurs).

Le site lui-même possède des fonctionnalités comme, la barre de recherche, les filtres, l'affichage en temps réel et image, information du tracé, connexion à un compte administrateur...

L'application mobile possède des fonctionnalités comme Bouton « Démarrer un tracé », Bouton « Stop », Tracé sensible, Point de départ et d'arrivée, Tracé téléchargeable, Historique...

4.2.1. Définir les différentes interfaces et leurs visibilité

Pour l'application mobile :

- **Menu principal** (menu par défaut lors de l'ouverture de l'application) :

- ◆ « *Démarrer un tracé* » : permet à l'utilisateur de commencer le tracé de son trajet
- ◆ « *Historique* » : voir ses précédents tracés
- **Option de tracé :**
 - ◆ « *Stop* » : permet d'arrêter le tracé
 - ◆ « *Télécharger* » : permet de télécharger l'image de son tracé
 - ◆ « *Enregistrer* » : permet d'enregistrer dans l'historique de l'application
 - ◆ Icône « *Accueil* » : retourner à l'accueil
- **Formulaire d'enregistrement** (menu par défaut lors de l'ouverture de l'application) :
 - ◆ Champ « *Date* » : entrer la date du tracé
 - ◆ Champ « *Mode de transport* » : permet de choisir le mode de transport
 - ◆ Champ « *Nom du tracé* » : entrer le nom du tracé
 - ◆ « *Valider* » : permet d'enregistrer les informations entrées dans le formulaire
 - ◆ Icône « *Accueil* » : retourner à l'accueil
- **Historique :**
 - ◆ « *Afficher les informations* » : nom du tracé, durée et mode de transport

- ◆ « *Supprimer* » : supprimer un tracé
- ◆ « *Favoris* » : mettre un tracé en favoris

Pour le site web :

- **Page d'accueil** (menu par défaut lors de l'ouverture de l'application) :
 - ◆ « *Connexion* » : permet à l'utilisateur de se connecter en administrateur
 - ◆ « *Barre de recherche* » : permet à l'utilisateur de rechercher d'anciens tracés
- **Liste des tracés** (point de vue de l'invité) :
 - ◆ « *Afficher* » : permet à l'utilisateur de visualiser un ancien tracé en temps réel / Image
 - ◆ « *Connexion* » : permet à l'utilisateur de se connecter en administrateur
 - ◆ Icône « *flèche* » : revenir à la page d'accueil
- **Liste des tracés** (point de vue de l'administrateur) :
 - ◆ « *Afficher* » : permet à l'utilisateur de visualiser un ancien tracé en temps réel (« En temps réel ») / Image (« En image »)
 - ◆ « *Corbeille* » : permet de supprimer ses tracés
 - ◆ « *Connexion* » : permet à l'utilisateur de se connecter en administrateur
 - ◆ Icône « *flèche* » : revenir à la page d'accueil

- **Affichage d'un tracé :**
 - ◆ « Afficher les informations » : visualiser les informations d'un tracé
 - ◆ « Connexion » : permet à l'utilisateur de se connecter en administrateur
 - ◆ Icône « *flèche* » : revenir à la page d'accueil
- **Page de connexion :**
 - ◆ Champ « *Email* » : indiquer l'email
 - ◆ Champ « *mot de passe* » : indiquer le mot de passe
 - ◆ « Connexion » – permet à l'utilisateur de se connecter en administrateur
 - ◆ Icône « *flèche* » : revenir à la page d'accueil

4.2.2. Décomposition en tâches

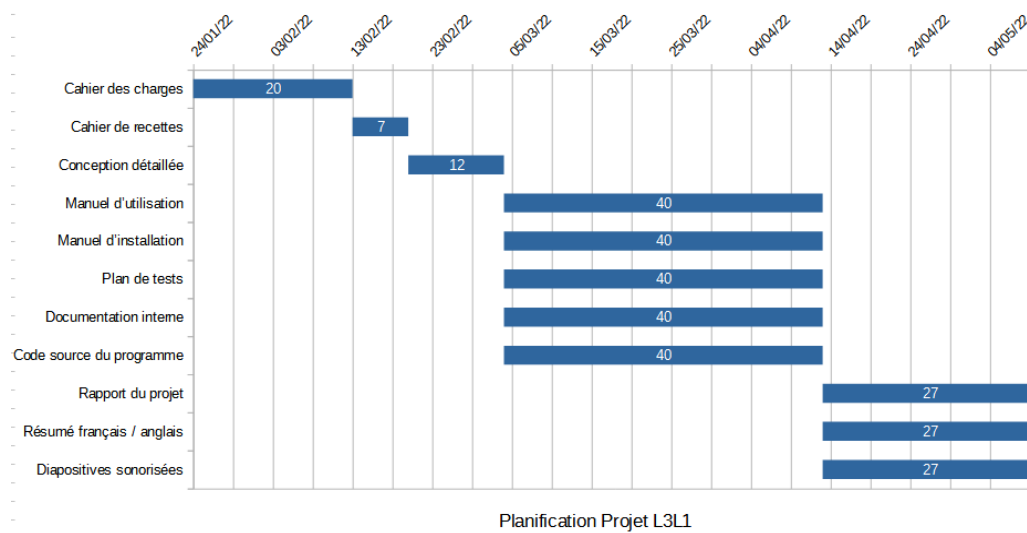


Illustration 8: Diagramme de Gantt

4.2.3. Méthodes et outils

Les outils de développement qui permettront de réaliser le projet sont :

- Les langages de programmation Java, Javascript, HTML, CSS, XML, SQL;
- Le SVN pour pouvoir gérer la synchronisation des différentes versions des fichiers sources.

4.2.4. Standard et outils

Les outils et le standard qui supporteront la documentation sont le plan de tests.

4.2.5. Références

- ◆ Le cahier des charges
- ◆ Le cahier de recettes
- ◆ Diagramme de déploiement
- ◆ Diagramme de cas d'utilisation
- ◆ Diagramme de classes
- ◆ Plan de développement

4.3. Les classes et librairies pour l'application

Dans cette partie nous allons présenter les classes qui seront utiles lors du développement de l'application.

Tout d'abord, Android Studio possède des classes par défaut (que nous allons utiliser) tels que :

- **Androidmanifest.xml** : ce fichier permettra de déclarer toutes les activités de l'application
- **MainActivity.java** : cette classe correspond à la partie principale du développement, elle permet la mise en place des fonctionnalités de l'application
- **activity_main.xml** : ce fichier permettra d'indiquer les composants graphiques qui devront s'afficher sur l'activité
- **colors.xml** : se situe dans le répertoire « *value* » (avec **strings.xml**), il permet de définir les couleurs de l'application
- **strings.xml** : se situe dans le répertoire « *value* » (avec **colors.xml**), il permet de regrouper tous les textes de l'application
- Il faudra autant de classes xml que de pages dans l'application pour indiquer tous les composants graphiques :
 - pour le menu principal
 - pour la page du tracé avec le bouton “Stop”
 - pour la page du tracé avec les deux boutons “Télécharger” et “Enregistrer”

- pour l'historique
- pour l'affichage des informations d'un tracé

Ensuite nous avons réfléchi à d'autres classes java qui nous paraissent essentielles au développement. Voici les classes :

- **Capteurs** : une classe qui permettra d'avoir accès aux capteurs (l'accéléromètre et le GPS) du téléphone
- **BaseDeDonnées** : correspond à une base de données qui contiendra tous les informations des anciens tracés
- **BDPositions** : correspond à une base de données qui contiendra toutes les positions possible en France
- **Tracé** : une classe qui permettra de faire le dessin en fonction des mouvements du téléphone
- **Les classes fragments** : ces classes java contiendront toutes les actions de chaque vue de l'application (*exemple* : la classe `FragmentHistorique` qui contiendra toutes les actions de l'historique, et qui sera associée à sa classe `.xml`). Il faudra faire autant de classes fragments, qu'il a de vues sur notre application

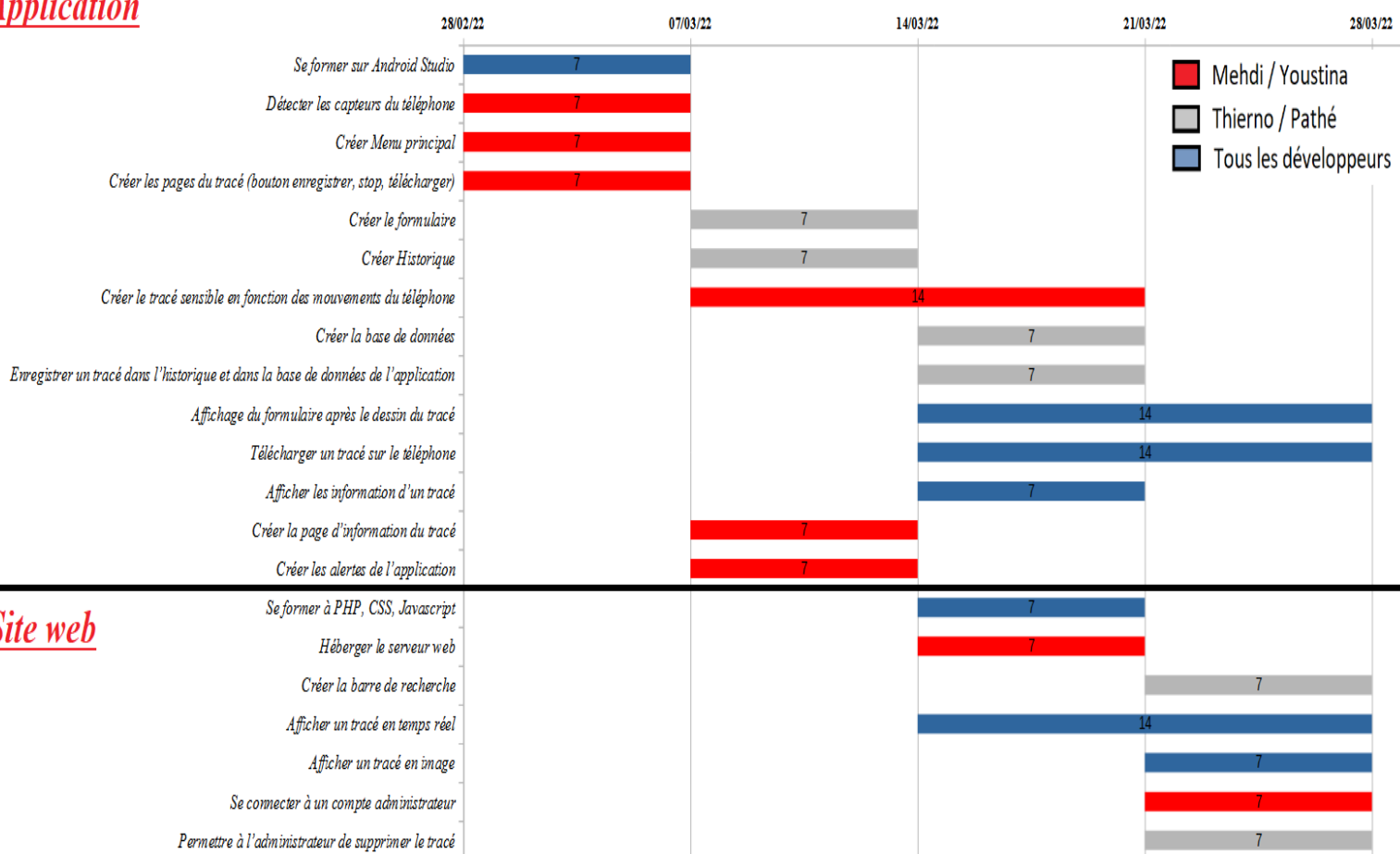
Pour finir, nous allons utiliser la librairie **SensorManager** qui va nous permettre d'utiliser les capteurs du téléphone notamment l'accéléromètre.

5. Annexes

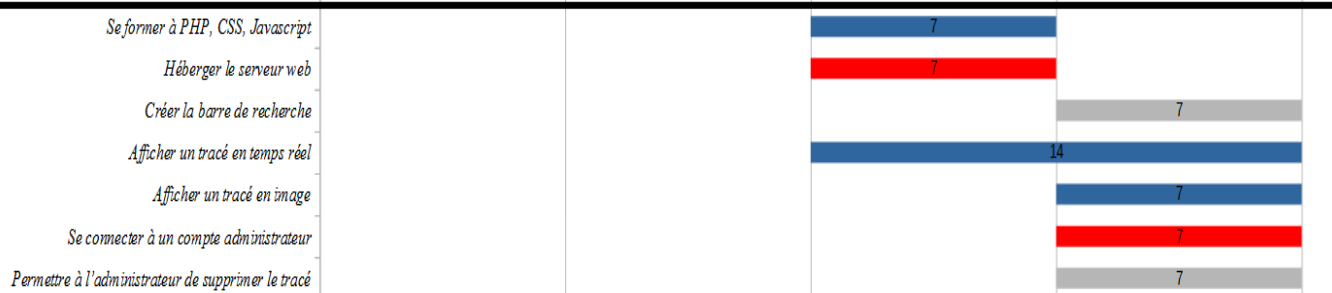
◆ Cahier des charges

◆ Cahier de recettes

Application



Site web



Planification Projet L3L1

Illustration 9: Diagramme de Gantt (Plan de développement)

6. Glossaire

- ★ **CONCEPTION DETAILLEE** : détermine les spécifications de chaque élément de l'équipement ou du processus
- ★ **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
- ★ **PORTAIL WEB** : Site web qui offre une porte d'entrée commune à un large éventail de ressources et de services accessibles sur Internet et centrés sur un domaine d'intérêt ou une communauté particulière
- ★ **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

7. Références

<https://www.ens.math-info.univ-paris5.fr/projets-informatiques/projets/documentation>

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

[https://developer.android.com/studio?](https://developer.android.com/studio?hl=fr&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuktzDN4F0N1KgZu1Tlems1pzjIfU67MhqFr7r6C5SGZH_JMTUi8yok8aAku9EALw_wcB&gclsrc=aw.ds)

[hl=fr&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuktzDN4F0N1KgZu1Tlems1pzjIfU67MhqFr7r6C5SGZH_JMTUi8yok8aAku9EALw_wcB&gclsrc=aw.ds](https://developer.android.com/studio?hl=fr&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuktzDN4F0N1KgZu1Tlems1pzjIfU67MhqFr7r6C5SGZH_JMTUi8yok8aAku9EALw_wcB&gclsrc=aw.ds)

[https://forge.ens.math-info.univ-paris5.fr/login?back_url=https%3A%2F](https://forge.ens.math-info.univ-paris5.fr/login?back_url=https%3A%2F%2Fforge.ens.math-info.univ-paris5.fr%2Fattachments%2Fdownload%2F4596%2Fpre%25cc%2581sentation2%2520Kourte%2520projet%2520tutore%25cc%2581.pdf)

[%2Fforge.ens.math-info.univ-paris5.fr%2Fattachments%2Fdownload%2F4596%2Fpre%25cc%2581sentation2%2520Kourte%2520projet%2520tutore%25cc%2581.pdf](https://forge.ens.math-info.univ-paris5.fr/login?back_url=https%3A%2F%2Fforge.ens.math-info.univ-paris5.fr%2Fattachments%2Fdownload%2F4596%2Fpre%25cc%2581sentation2%2520Kourte%2520projet%2520tutore%25cc%2581.pdf)

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

8. Index

Index des figures

| | |
|---|-----------|
| <i>Illustration 1: Image de présentation</i> | <i>1</i> |
| <i>Illustration 2: Diagramme de déploiement</i> | <i>9</i> |
| <i>Illustration 3: Diagramme de cas d'utilisation (application)</i> | <i>10</i> |
| <i>Illustration 4: Diagramme de cas d'utilisation (site web)</i> | <i>11</i> |
| <i>Illustration 5: Diagramme de classes (Application)</i> | <i>12</i> |
| <i>Illustration 6: Diagramme de classes (Site web)</i> | <i>13</i> |
| <i>Illustration 7: Modèle cycle de vie en V</i> | <i>14</i> |
| <i>Illustration 8: Diagramme de Gantt</i> | <i>18</i> |
| <i>Illustration 9: Diagramme de Gantt (Plan de développement)</i> | <i>22</i> |

Index lexical

| | |
|--------------------|--------|
| Android | 15, 23 |
| ANDROID | 23 |
| Android Studio | 8 |
| application mobile | 5, 15 |
| APPLICATION MOBILE | 23 |

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

| | |
|--------------------------------|----------|
| architecture | 5, 8 |
| Architecture | 8 |
| artefacts | 9 |
| associations | 9 |
| attributs | 12 |
| barre de recherche | 15 |
| Barre de recherche | 17 |
| classes | 5, 7, 12 |
| composants | 6 sqq. |
| conception détaillée | 5 sq. |
| Conception détaillée | 1 |
| conception générale | 5 sq. |
| connexion | 15, 18 |
| Connexion | 17 sq. |
| CSS | 19 |
| Diagramme de cas d'utilisation | 10, 19 |
| diagramme de classes | 12 |

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

| | |
|---------------------------------|----------|
| Diagramme de classes | 12, 19 |
| diagramme de déploiement | 8 sq. |
| Diagramme de déploiement | 19 |
| diagrammes de cas d'utilisation | 10 |
| filtres | 15 |
| fonctionnalités | 15 |
| fonctions | 5, 7, 12 |
| HTML | 19 |
| image | 15 sqq. |
| Image | 17 |
| information du tracé | 15 |
| Java | 8, 19 |
| Javascript | 19 |
| JavaScript Object Notation | 8 |
| librairies | 5 |
| méthodes | 5, 7 |
| Méthodes | 19 |

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

| | |
|-----------------------|-----------|
| modèle de cycle en V | 14 |
| MySQL | 8 |
| nœuds | 9 |
| paquetages | 5 |
| PHP | 8 |
| Plan de développement | 19 |
| projet | 7, 15, 24 |
| Projet | 1 |
| projet sont | 19 |
| site web | 5, 15, 17 |
| Site web | 23 |
| SQL | 19 |
| SVN | 19 |
| temps réel | 15, 17 |
| UML | 12 |