|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Informaticien/-ne CFC**  Travail pratique individuel 2024 (TPI) |  | |

Nom du candidat : Elvin Kuci

**« Road Traffic Simulator »**

### Sommaire

Résumé du rapport du TPI 2

1 Les grandes lignes du projet 3

1.1 Analyse de la situation initiale 3

1.2 Analyse de l’état désiré 3

1.3 Cahier des charges / exigences du système 3

1.4 Organisation du projet 3

2 Travaux préparatoires 5

2.1 Installation de l’environnement 5

2.2 Création d'un projet JavaFX 5

2.3 JavaFX-TT 6

2.4 Moving-TT 8

3 Analyse 11

3.1 Analyse de l’état désiré 11

3.2 Variantes 13

3.3 Sécurité de l’information et protection des données 15

4 Conception 16

4.1 Exigences du système 16

4.2 Architecture du système 16

4.3 Maquettes 16

4.4 Concept d‘implémentation 17

4.5 Concept de tests 19

5 Réalisation 20

5.1 Spécifications détaillées 20

5.2 Design du système 20

5.3 Configuration xyz 20

6 Test 21

6.1 Procédure de test 21

6.2 Protocol de test 21

6.3 Signature du protocole de test 21

7 Conclusion 22

7.1 Améliorations possibles 22

7.2 Auto-évaluation 22

8 Bibliographie: liste des sources et références 23

9 Glossaire 24

10 Signatures 25

11 Annexes 26

# Résumé du rapport du TPI

Le Résumé du rapport du TPI est une présentation conceptuelle du travail effectué et du résultat attendu d’au maximum une page A4 qui permettent au lecteur une rapide compréhension du rapport de travail. . Il contient trois paragraphes : Situation de départ, mise en œuvre, résultats.

Pour plus de détails, veuillez vous référez au Manuel ICT - partie B : Documentation / rapport du TPI - Question 14 - page 81.

# Les grandes lignes du projet

## Analyse de la situation initiale

Quelle est la situation actuelle? Vous pouvez éventuellement illustrer le processus.

Cette application n’a pas de situation initiale. Ce mandat est le premier mandat sur ce projet et tout est à faire depuis zéro.

## Analyse de l’état désiré

A quoi doit ressembler la solution/le produit fini? Vous pouvez éventuellement illustrer le nouveau processus.

L’objectif final est d’avoir une application pour ordinateur permettant de lancer une simulation d’un carrefour. Il sera possible de modifier et sauvegarder de nombreux paramètres telle que les feux, la vitesse de la route, etc… Cette application permettra de tester et visualiser un ensemble de paramètres appliqué sur un carrefour.

## Cahier des charges / exigences du système

Cahier des charges détaillé découlant du mandat de projet. Définition exacte et mesurable des objectifs à atteindre.

Décrire et structurer les exigences du système désiré.

L’application devra gérer les 4 tronçons de routes, à circulation bidirectionnelle, lié par un carrefour équipé de feux se signalisation. Une extension possible devra être la simulation de ces mêmes tronçons de route avec un rond-point. Ce développement ne devra pas être réalisé, mais pris en compte dans l’analyse et la conception, afin que l’application puisse être modifiée facilement.

L’objectif est aussi de fournir à l’utilisateur une application paramétrable pour agir sur le résultat visuel de la simulation :

* Échelle pour la représentation graphique en pixel par mètres
* Limitation de la vitesse, distance des et densité du trafic
* Temps de réaction, accélération, décélération et temps de sécurité
* Temps des feux et ordre de ceux-ci
* Sauvegarde du jeu de paramètre sous un nom

## Organisation du projet

Citer la méthode de gestion de projet utilisée (Méthode en phases, agile, scrum, …)

Etablir la liste des participants au projet avec leur rôle respectif.

Méthodologie en cascade / séquenciel

Support : planning + journal de travail (blablater dessus)

Décrire comment sont gérées les sauvegardes de la documentation et du code (fréquence, supports utilisés, utilisation de logiciel de gestion de version, …)

*TODO : Check methode with best BUDDY*

Méthode: Séquenciel ? Incrémental ? Cascade ?

Additionellement, voici la liste des participants au projet :

|  |  |
| --- | --- |
| Participant | Rôle |
| Kuci Elvin | Candidat, développeur de l’application |
| Bütschi Daniel | Supérieur professionnel |
| Hertling Frédéric | Formateur en entreprise |
| Perroud Didier | Expert principal |
| Thomet Samuel | Expert secondaire |

### Sauvegarde et préservation

Le code et la documentation sont présent dans le Git du projet. Un commit est émis à chaque modification d’un élément dans le code et/ou la documentation.

Chaque groupe de commit majeur est édité sur sa propre branche et est merge sur la branche « main » une fois que le travail sur celle-ci est terminé.

Une fois que l’application atteint un objectif et est stable, une branche « version\_x.y.z » est créer. Cette branche sert de backup pour rapidement revenir à la dernière version stable.

Pour revenir à un ancien commit, exécutez un « git log » pour voir l’identifiant d’un commit et un « git reset IdDuCommit » pour revenir à l’ancien commit.

Bien entendu, les commits sont journalièrement pousser vers un repository distant. Dans le cadre de ce projet, la repository initiale est <https://github.com/alikadev/RoadTrafficSimulator>. Ce repository est un repository personnelle sur lequel j’ai push les premiers commits. De plus, tous le code est envoyé vers <https://gitlab.emf-infopro.ch/EMF-Informatique/tpi/tpi-2024-roadtrafficsimulation> qui est le repository officiel.

Pour aller sur une branche, faites un « git switch NomDeLaBranche ». Si vous n’avez pas la branche en local, faites un « git fetch --all » suivi d’un « git checkout NomDeLaBrancheDistante –detach ».

*Ajout des exemples, images, et printscreen*

# Travaux préparatoires

## Mise en place d’un projet JavaFX

Pour ce projet, NetBeans va être utilisé comme IDE. Vous pouvez installer NetBeans depuis le [site web officiel](https://netbeans.apache.org/front/main/download/).

Ensuite, pour créer un projet JavaFX, Allez dans « File > New Projet… » Et créer un projet « Java with Maven > FXML JavaFX Maven Archetype (Gluon) ». Maven va installer automatiquement JavaFX et mettre en place un projet de base. Vous pouvez ensuite lancer l’application et tout devrait fonctionner.

La raison pour laquelle nous allons utiliser Maven est car l’installation de JavaFX sera faite automatiquement. Additionnement, beaucoup plus de documentation est écrite sur les projets Maven JavaFX que Ant. Notez que l’architecture arm64 n’est pas supportée par Maven - JavaFX. JavaFX n’a pas un bon support de cette architecture et NetBeans causera de nombreuses difficultés durant la mise en place de l’application.

Vous pouvez aussi remarquer que Maven ajoute 1 fichier important : « module-info.java ». Ce fichier défini le module (l’application) et ses exports, imports, etc…

## Création d'un projet JavaFX

Le test technologique implémenté a pour but de tester la communication entre le Thread Java et le Thread JavaFX. Voici la liste des technologies qui vont être tester :

* Échange de données entre le Thread JavaFX et le Main Thread
* Expérimentation de Platform.runLater
* Expérimentation de Binding
* Expérimentation de Task

Le choix de sa technologie n’a pas été fait au hasard. La communication entre le « backend (Worker) » et le « frontend (JavaFX) » est une des parties les plus complexe de ce projet.

### Structure MVC avec JavaFX

|  |  |
| --- | --- |
| Le projet est nommé JavaFX-TT et sera une application MVC (Model/Worker – View – Controller) simple.  Les Views sont dans le src/main/ressources/app/views au format FXML. Leur location n’est pas aléatoire, « src/main/ressources » est le dosser depuis lequel les ressources de l’application sont copié avec Maven. Vu que les FXML ne sont pas compiler, ce sont des ressources.  Le contrôleur se trouve dans « app/controller » et le Worker se trouve dans « app/workers ».  (Figure 1 : le projet sur NetBeans) | A screenshot of a computer  Description automatically generated |

## JavaFX-TT

### Fonctionnement

|  |  |
| --- | --- |
| Quand vous lancez l’application, vous allez voir un champ texte qui est lié à la « value » du label. Une fois que vous appuyez sur « Fetch data », le « value » affiche « Fetching data… » pendant 1 seconde et va charger des valeurs dans le tableau à droite.  Ce test n’est pas très utile mais permet de bien tester le fonctionnement des différentes technologies utile pour le projet | A screenshot of a computer  Description automatically generated |
| A screenshot of a computer  Description automatically generated |

### Points importants de l’implémentation

Je ne vais pas écrire tout le code de l’application ci-dessous mais uniquement les parties intéressantes.

/\*\*

\* Update the list content

\* @param values The list of values to put in the list

\*/

public void updateList(List<String> values) {

// Use of Platform.runLater because it will be called by an external thread.

Platform.runLater(() -> listTask.getItems().setAll(values));

}

Lors de la mise à jour de la liste de l’application, l’utilisation du Platform.runLater permet à un second thread de modifier le contenu de la liste. Sans ce Platform.runLater, la mise à jour de la liste pourrai créer un java.lang.IllegalStateException. Cette erreur n’apparait pas à chaque clique mais arrivera au bout d’un moment.

/\*\*

\* Bind the label to the input and enable the buttonRunTask.

\*/

private void bindLabelToInputText() {

label.textProperty().bind(textLabel.textProperty());

buttonRunTask.setDisable(false);

}

/\*\*

\* Unbind the label from the input and disable the buttonRunTask.

\*/

private void unbindLabelToInputText() {

label.textProperty().unbind();

buttonRunTask.setDisable(true);

}

Ces 2 fonctions créer/retire le lien entre le texte du label « value » avec le champ texte (textLabel). Sans cette fonctionnalisée, un autre moyen serai d’utiliser un callback ce qui serai beaucoup moins efficient et dure à gérer.

// Create a task to set the new string value

Task<Void> fetchTask = new Task() {

@Override

protected Void call() throws Exception {

// Simulate a long operation (1 second)

wrk.fetchData();

return null;

}

};

// On task success, rebind the text

fetchTask.setOnSucceeded((WorkerStateEvent e) -> {

bindLabelToInputText();

});

// Start the task

new Thread(fetchTask).start();

La création d’une tâche (Task) dans ce cas permet d’ajouter des callbacks pour l’exécution de la tâche. Ici, après que le Worker à récupérer les données, le texte est lié au champ de texte. Comme vous pouvez le voir, la tâche est lancée via la création et exécution d’un nouveau thread.

### Récapitulatifs

* Platform.runLater permet d’interagir avec le thread JavaFX.
* Les Bindings permettent de lier les valeurs avec entre elle.
* Les Task permettent une gestion avancée des Threads et un meilleur contrôle de flux.

## Moving-TT

Ce test technologique a pour but de tester le déplacement d’image sur l’écran. Ceci permet de mieux comprendre comme créer des animations et simulations sur le « backend » et les afficher sur le frontend.

### Fonctionnement

|  |  |
| --- | --- |
| Quand vous lancerez cette application, vous verrez une flèche tourner autour du centre de la fenêtre.  Cette application n’a aucune action intéressante mais permet de mieux comprendre comment JavaFX gère ses entités et comment les déplacer sur l’écran. | A screen shot of a computer  Description automatically generated |

### Points importants du code

/\*\*

\* Set the arrow position.

\*/

public void setArrowPosition(double x, double y) {

arrow.setLayoutX(x);

arrow.setLayoutY(y);

}

/\*\*

\* Set the arrow rotation in degree.

\*/

public void setArrowRotation(double angle) {

arrow.setRotate(angle);

}

Ceci est le code permettant de déplacer et tourer une ImageView nommée arrow. La modification de ces valeurs ne demande pas de Platform.runLater.

// Update position

// sin(x) = SIN from -1 to 1

// sin(x)+1 = SIN from 0 to 2

// (sin(x)+1)/2 = SIN from 0 to 1

// (sin(x)+1)/2\*n = SIN from 0 to N

double angle = start \* Math.PI \* 2 / SEC\_PER\_ROT;

double xMult = ctrl.getLayerWidth() - ctrl.getArrowWidth();

double yMult = ctrl.getLayerHeight() - ctrl.getArrowHeight();

x = (Math.sin(angle) + 1) / 2 \* xMult;

y = (Math.cos(angle) + 1) / 2 \* yMult;

ctrl.setArrowPosition(x, y);

ctrl.setArrowRotation(-angle / Math.PI / 2 \* 360);

Et voici comment le Worker modifie la position de la flèche. Ce n’est rien de plus qu’un peu de trigonométrie.

// Sleep to reach a FPS goal

double now = getTime();

long sleepMS = Math.round((1.0/FPS - (now - start)) \* 1000);

if (sleepMS > 0) {

try {

Thread.sleep(sleepMS);

} catch (InterruptedException e) {

System.err.println("Unreachable");

}

}

Et voici comment le nombre d’image par second est gérer.

### Récapitulatifs

* ImageView.setLayoutX/Y(double) : Modifie la position d’une image
* ImageView.setRotate(double) : Modifie la direction de l’image en degré

# Analyse

L’analyse du projet à été réalisé le premier jour de projet. Cette phase permet de visualiser les charges de travail ainsi que la direction générale du projet.

## Analyse de l’état désiré

|  |  |
| --- | --- |
| L’objectif du produit fini sera de simuler et visualiser le trafic d’un carrefour à 4 voies en fonctions des différents paramètres éditables. L’application final aura une interface intuitive permettant à l’utilisateur de comprendre visuellement en temps réel les effets de différents paramètres. Il sera ainsi en mesure d’optimiser la séquence des feux de signalisation.  Comme vous pouvez le voir sur la figure ci-présente, le diagramme de cas d’utilisation :  Un « visiteur » peut se connecter et créer un compte.  Un « utilisateur » connecté peut exécuter et visualiser la simulation, modifier et sauvegarder des jeux de paramètres et se déconnecter.  Et la base donnée ne travaille qu’avec le compte et les paramètres. | A screenshot of a computer  Description automatically generated |

### Fonctionnement de la connexion

|  |  |
| --- | --- |
| Quand un visiteur arrive sur l’application, il sera invité à se connecter. Pour se faire, il va entrer son nom de compte, son mot de passe et appuyer sur « Se Connecter ».  Une fois que ceci est fait, l’application dois dévifier que les champs sont bel et bien entré.  Ensuite, l’application va récupérer le compte avec le même nom dans la base de donnée et vérifier si le mot de passe est bon. Si tout se passe bien, l’appliciation changera se vue pour entrer dans le mode de simulation. | A diagram of a computer  Description automatically generated |

### Création d’un jeu de réglage

|  |  |
| --- | --- |
| Lors de la création d’un jeu de réglage, l’utilisateur devra entrer le nom du jeu de réglage et appuyer sur « Créer le jeu de réglage ».  À ce moment, l’application vérifiera l’argument et récupèrera les réglages en cours.  Une fois cela fait, le jeu de réglage sera envoyé à la base de donnée et affichera le résultat de l’action. | A diagram of a diagram  Description automatically generated |

### Sauvegarde de paramètres - détails

|  |  |
| --- | --- |
| L’utilisateur n’est nullement obligé de créer un nouveau jeu de réglage à chaque modification. Il peut aussi sauvegarder de nouveaux réglages sur un jeu de réglage préexistant.  Le diagramme d’activité ci-présent démontre les deux cas et leurs fonctionnements.  Si l’utilisateur souhaite sauvegarder un jeu de réglages, il doit sélectionner le jeu dans la liste et cliquer sur sauvegarder.  Par contre, si l’utilisateur souhaite créer un nouveau jeu de réglages, il devra entrer le nom du jeu de réglage et cliquer sur créer.  Bien qu’il y ait 2 chemins différents dans ce diagramme, nous pouvons remarquer qu’ils ont un fonctionnement assez proche. Il sera donc probablement possible d’utiliser les mêmes fonctionnalités dans le code pour ces deux chemins ! | A diagram of a process  Description automatically generated |

### Exécution de la simulation

|  |  |
| --- | --- |
| Pour débuter la simulation, l’utilisateur n’aura qu’à appuyer sur « Commencer ».  Ensuite, l’application cachera le menu d’édition et démarrera le thread de la simulation.  Celui-ci s’occupera de déplacer et créer les véhicules ainsi que d’afficher le nouvel état et vérifier s’il doit arrêter.  Quand l’utilisateur appuyer sur « Arrêter », la simulation s’arrêtera. | A diagram of a computer  Description automatically generated |

## Variantes

Lors de la conception des structures de données et des flux de l’application, 2 méthode de gestion des données intéressantes me sont venu en tête. La première est plus simple mais moins facile à généraliser et la deuxième est plus complexe mais simple a généraliser.

### Variante 1

enum RoadType =

STRAIGHT\_ROAD

CROSSROAD

type Road:

getType(): RoadType

class StraightRoad impl Road;

getType(): RoadType = return STRAIGHT\_ROAD

// ...

class Crossoad impl Road:

getType(): RoadType = return CROSSROAD

// ...

circuit: List<Road> = { ... }

for rd in circuit:

if rd.getType() == CROSSROAD:

rd = (Crossroad) rd;

// Show crossroad

elif rd.getType() == STRAIGHT\_ROAD:

rd = (StraightRoad) rd;

// Show straight\_road

RoadType est une constante permettant de différencier entre les différents types de routes.

Road est le type parent permettant de définir que les routes doivent implémenter getType.

StraightRoad et Crossroad sont des exemples de routes. Ils ont des propriétés différentes.

Donc dès qu’on utilise le circuit, nous devons récupérer le type de route et la ré-interprété en le bon type de route.

Cette solution est très simple à implémenter mais peut devenir très dure à agrandir. Dès le moment où nous voulons ajouter un nouveau type de route, nous devons ajouter une énumération, une classe et modifier tous les if et switch-cases.

### Variante 2

type Road:

getProps(): List<Pair<Label, Props>>

class StraightRoad impl Road:

getProps(): List<Pair<Label, Props>> =

return // Road props here...

class Crossroad impl Road:

getProps(): List<Pair<Label, Props>> =

return // Crossroad props here ...

circuit: List<Road> = { ... }

for rd in circuit:

(label, property) = rd.getProps()

JFX\_HBox hbox = new JFX\_HBox()

hbox.add(new JFX\_Label(label))

hbox.add(new JFX\_TextField().bindValue(property))

editMenu.add(hbox)

Road est le type parent permettant de définir qu’une route implémentera getProps. Cette méthode est utilisée pour récupérer les propriétés d’une route.

StraightRoad and Crossroad sont des types de routes différent.

Dès le moment où nous devons afficher les champs permettant d’éditer une route, nous avons juste à récupérer les propriétés et les afficher. Le fait que ce soit une Crossroad, une StraightRoad ou un chemin de terre ne change rien, le code devient totalement réutilisable

### Choix de variante

Pour décider quelle variante utiliser, nous pouvons faire une liste des avantages et inconvenants avec une pondération et faire la somme des points de chaque variante. Plus la somme est haute, moins l’idée est intéressante dans ce cadre. Pour cette matrice, j’ai choisi 3 points majeurs :

* La difficulté initiale : Difficulté de la première implémentation

Ce n’est pas vraiment un problème au long terme, mais ça risque de rendre la première implémentation plus longue et fastidieuse. La pondération est donc de 1.

* Difficulté future : Difficulté d’agrandissement de la solution dans le futur.

C’est très important dans le cadre de ce mandat de faire une application qui sois extensible et améliorable. La pondération est donc de 3.

* Risque de bug : Risque de commettre une erreur et difficulté de debug

Le risque de bug peut causer des problèmes ennuyants et prendre du temps mais ce n’est en aucun cas autant important que l’extensibilité de l’application. Une pondération de 2 me semble honnête.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pondération | Variante 1 | Variante 2 |
| Difficulté initiale | x1 (au début) | 1 (simple) | 3 (complexe) |
| Difficulté future | x3 (extensible) | 4 (immuable) | 2 (constante – se facilite) |
| Risque de bug | x2 (risque + debug) | 2 (normal) | 1 (faible ou simple débugger) |
| Résultats |  | 1 + 12 + 4 = **17** | 3 + 6 + 2 = **11** |

Malgré les quelques mots clé que j’ai mis dans la matrice ci-dessus, je pense que c’est une bonne idée de repasser sur chaque point ci-dessous. Mise à part les détails, nous pouvons voire que la deuxième variante est beaucoup plus intéressante dans ce cas !

#### Difficulté initiale

La première variante est probablement la plus simple possible. Une énumération et un « switch-case » permet d’implémenter cette variante. Une difficulté de 1 me parait bon.

La deuxième variante est plus complexe. L’abstraction du type de route, et même de ce qu’est une route en premier lieu rendra la première implémentation complexe. Au lieu de remplacer des bindings sur des onglets préfabriquer, il faudra générer les liens entre les objets manuellement. Une difficulté de 3 me semble correcte !

#### Difficulté future

La première option à un gros problème sur ce point. À chaque fois que nous voudrons ajouter un type de route, de devrons passer à travers tous le code et ajouter une condition pour ce nouveau type de route. C’est très facile d’oublier une condition et de causer une erreur qui peut être très difficile à trouver.

La deuxième option a un gros avantage sur ce point : Vu qu’une route est un concept abstrait de base, l’ajout d’un type de route ne consiste qu’à implémenter la route et la rajouter dans la simulation. De plus, au fil du développement, de plus en plus de type de route seront ajouter dans le code et l’ajout de nouvelles routes ne deviendra qu’un copier-coller de paterne qui ont déjà été créer auparavant. 2 est la difficulté que j’ai choisie.

#### Risque de bug

La première option est relativement standard. Il y a un risque de bug typique pour une application de cette nature et les résoudre ne sera pas trop complexe non plus. De simple solution sont simple à debugger. Le risque est de 2.

La deuxième option est encore plus simple à debugger. S’il manque un champ pour une route, c’est qu’il n’est pas retourné ou qu’il y a un problème d’affichage. C’est vraiment aussi simple que ça, 1 me semble correcte !

## Sécurité de l’information et protection des données

Cette application subit un problème que toutes les applications avec la même fonctionnalisée ont : ils sauvegardent les identifiants et mots de passe de leurs utilisateurs.

Une personne malicieuse pourrait réussir à s’introduire dans la base de données et récupérer les mots de passe des utilisateurs. Le véritable risque est que certains utilisateurs vont utiliser les mêmes identifiants et mots de passe pour d’autres comptes.

Pour régler ce problème, nous allons alors hasher les mots de passes pour qu’ils ne puissent pas être retrouvé. Le hashage est un procédé informatique qui consiste à récupérer un texte en entré et à un sortir une emprunte unique. Le point intéressant est que la fonction de hash va toujours générer la même emprunte et le texte original ne peut pas être retrouvé avec l’emprunte. Nous pourrons alors comparer les empruntes sans divulguer le mot de passe de l’utilisateur !

# Conception

## Exigences du système

Les tronçons ainsi que le carrefour possèderont des identifiants unique et un système de connexion numérotées. La construction logique et visuelle se fera par le développeur mais dois être pensée générique.

Additionnellement, une gestion du flux et de la densité des véhicules par minutes pourra être affecté au tronçon. Le véhicule sera créé avec tous ses paramètres associés ainsi qu’une légère variabilité dans une plage de tolérance permettant de mieux simuler une route. Les véhicules réagiront aux comportements du véhicule suivant en essayant de maintenir une distance de sécurité.

De plus, l’utilisateur doit pouvoir se logger à sa session sur différents PC où l’application est installé et retrouver ses jeux de paramètres. Par conséquent, les données seront stockées sur une base de données hébergée sur emf-infopro-tpi.ch.

Finalement, l’interface graphique représentera de manière simple les tronçons de route et le carrefour avec les feux. L’accès aux paramètres devra être pensé dans une approche ergonomique et les éléments dynamique seront les véhicules et la couleur des feux de signalisation.

## Architecture du système

La structure du projet est relativement simple. L’application Java communique directement avec la base de données SQL distante.

### L’application cliente

L’application cliente est développée en Java avec l’utilisation de la librairie graphique JavaFX. JavaFX est une bibliothèque de code développé par Oracle permettant de créer des interface utilisateur avec Java.

## La base de données

TODO : Filll info about the database

## Maquettes

La maquette ci-dessous est la maquette de l’application. Elle permet aussi de représenter le flux de l’application à travers les différentes cliques.

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated | Tout commence en haut à gauche avec la page de connexion qui permet de se connecter.  Ensuite, la page de simulation permet d’éditer les paramètres généraux, gérer les jeux de réglages. Quand vous cliquez sur une route ou un carrefour, vous verrez ses paramètres que vous pourrez éditer.  Les carrefours ont aussi un onglet « Feux » permettant de controller la signalisation. Vous pourrez donc controller les feux de chaque direction ainsi que le temps au vert, orange et quels feux doivent être mis au vert quand celui-ci passe au rouge. |

## Diagramme de classe

Le diagramme de classe ci-dessous est celui qui a été créer durant la conception. Allez au point 6 pour retrouver le dernier diagramme de classe.

|  |  |
| --- | --- |
| A diagram of a software company  Description automatically generated with medium confidence | L’application est composée de contrôleurs (en vert) implémentant la même interface pour que le Worker (en bleu) puisse communiquer avec chaque contrôleur. De plus, les beans (en magenta) représente les différentes structures qui seront utilisé par l’application.  Le SimulationCtrl est le contrôleur de la page de simulation et le LoginCtrl est contrôleur de la page de login.  Les vues ne sont pas présentées sur ce schéma car ce sont des FXML et non des classes. |

## Diagrammes de séquence d’interaction

### Se connecter

|  |  |
| --- | --- |
| TODO : Write about this | A screenshot of a computer  Description automatically generated |

### Sauvegarder les paramètres

|  |  |
| --- | --- |
| TODO : Complete this | A screenshot of a diagram  Description automatically generated |

### Exécuter la simulation

|  |  |
| --- | --- |
| TODO : Complete this | A screenshot of a computer  Description automatically generated |

## Concept de tests

Les tests qui seront réalisés sont des tests dit « Blackbox ». Ceci signifie que ce sont des tests que nous pouvons effectuer sans connaitre la structure interne de la solution / de l’application.

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Objet | Description du test | Résultat attendu |
| Login | Créer un compte | Créer un compte qui n’existe pas dans la base de données. | Le compte sera créé.  La page change sur « Simulation ». |
| Créer un compte qui existe déjà. | Le compte ne sera pas créé.  Afficher une erreur. |
| Créer un compte avec des champs vide. | Afficher une erreur. |
| Créer un compte avec le champ « confirmation de mot de passe » qui est diffèrent du « mot de passe » | Afficher une erreur. |
| Se logger | Se logger à un compte qui n’existe pas | Afficher une erreur. |
| Se logger à un compte qui existe avec le mauvais mot de passe | Afficher une erreur. |
| Se logger à un compte qui existe avec le bon mot de passe | La connection va fonctionner.  La page change sur « Simulation ». |
| Simulation | Charger un jeu de réglages | Charger un jeu de réglages sans sélectionner un jeu | Afficher une erreur. |
| Charger un jeu de réglages | Charger le jeu et afficher le succès |
| Sauvegarder un jeu de réglages | Sauvegarder un jeu de réglages sans sélectionner un jeu | Afficher une erreur. |
| Sauvegarder un jeu de réglages | Sauvegarder dans la base donnée et afficher le succès |
| Créer un jeu de réglages | Créer un jeu de réglages sans nom | Afficher une erreur. |
| Créer un jeu de réglages avec un nom qui existe déjà | Afficher une erreur. |
| Créer un jeu de réglages avec un nom qui existe déjà mais qui a été ajouté après le lancement de l’application. Le jeu est donc dans la base de données mais pas la liste. | Afficher une erreur. |
| Créer un nouveau jeu de réglages | Créer le jeu de réglages et afficher le succès |
| Simulation | Lancer la simulation avec les réglages de départ | Le menu d’édition se ferme et la simulation fonctionne. |
| Quitter la simulation. | La simulation s’arrête et le menu d’édition s’ouvre. |
| Une fois que la simulation c’est stabilitée, vérifiez les statistiques sur la simulation. | La densité de la route équivaut au « In » et est le |
| Réglage général  Pixels par mètre | Changer le « Pixels par mètre » à une valeur inférieur à 0. | Afficher une erreur. |
| Changer le « Pixels par mètre » à une valeur entre 1 et 100. | Change la taille et fonctionne normalement. |
| Réglage général  Facteur de vitesse | Modifier le « Facteur de vitesse » à une valeur inférieur à 0. | Afficher une erreur. |
| Modifier le « Facteur de vitesse » à une valeur entre 0.2 et 10. Une valeur plus grande pourrait causer un problème dans tous les cas. | Changer le facteur de vitesse. |
| Réglage véhicule  Décélération | Modifier cette valeur. | La valeur cause un changement lors des décélérations standard. |
| Réglage véhicule  Décélération % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage véhicule  Décélération d’urgence | Modifier cette valeur. | La valeur cause un changement lors des décélérations d’urgence. |
| Réglage véhicule  Décélération  d’urgence % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage véhicule  Accélération | Modifier cette valeur. | La valeur cause un changement lors des décélérations d’urgence. |
| Réglage véhicule  Accélération % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage véhicule  Temps de sécurité | Modifier cette valeur. | La valeur cause un changement de distance entre les véhicules |
| Réglage véhicule  Temps de sécurité % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage véhicule  Temps de réaction | Modifier cette valeur. | La valeur cause un changement de temps de réaction |
| Réglage véhicule  Temps de réaction % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Cliquer sur une route | Nous cliquons sur un tronçon de route standard | Le menu de la route s’ouvre. |
| Cliquer sur un carrefour | Nous cliquons sur un carrefour. | Le menu de carrefour et le menu de feux de signalisation d’ouvre |
| Réglage route  Limitation de vitesse | Modifier cette valeur. | La valeur cause un changement de comportement des véhicules. |
| Réglage route  Limitation de vitesse % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage route  Taille | Modifier cette valeur. | La valeur cause un changement visuel ainsi qu’un changement de comportement des véhicules. |
| Réglage route  Densité | Modifier cette valeur. | La valeur cause un changement de la vitesse d’apparition des véhicules |
| Réglage route  Densité % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglage carrefour  Limitation de  vitesse % | Modifier cette valeur. | La valeur cause un changement de comportement des véhicules. |
| Réglage carrefour  Limitation de  vitesse % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. |
| Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. |
| Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. |
| Réglages feux  Temps vert | Modifie cette valeur avec une valeur inférieur à 0. À tester sur les 4 différents feux. | Afficher une erreur. |
| Modifie cette valeur avec une valeur différente de 0. À tester sur les 4 différents feux. | La valeur cause un changement de temps de vert à feux tester. |
| Réglages feux  Temps orange | Modifie cette valeur avec une valeur inférieur à 0. À tester sur les 4 différents feux. | Afficher une erreur. |
| Modifie cette valeur avec une valeur différente de 0. À tester sur les 4 différents feux. | A valeur cause un changement de temps d’orange à feux tester. |
| Réglages feux  Passage au rouge | Modifie les « croix ». À tester pour toutes les lignes et tous les feux. | Lors d’un passage au rouge, les lignes sélectionnées passent au vert |
| Réglages feux  Prochain | Appuyer sur le bouton « Prochain » | Un changement de feu en cours s’opère dans une direction constante (horaire ou anti-horaire, pas de hasard !) |

# Réalisation

## Spécifications détaillées

## Design du système

## Configuration xyz

Conseil: D’autres chapitres peuvent s’ajouter à cette partie en fonction du projet.

# Différence entre la conception et la réalisation

Trouver des choses à dire…

# Test

## Procédure de test

Pour tester l’application, j’ai lancer l’application sur ma machine professionnel (x86-64 Windows 10) depuis Intellij IDEA Community Edition et j’ai réalisé les tests un par un en suivant les instructions. Pour reproduire ces tests, il vous faudra une machine (x86-64, Windows 10 / Debian) ainsi qu’une connexion réseau.

L’application ne fonctionnera pas sur l’architecture ARM car JavaFX ne fonctionne que sur x86-64. Je n’ai pas pu tester l’application sur un Mac (x86-64) car je n’en n’ai pas sous la main.

## Protocol de test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Objet testé** | **Description du test** | **Attente** | **Heure** | **Résultat** | **Visa** |
| 1 | Créer un compte | Créer un compte qui n’existe pas dans la base de données. | Le compte sera créé.  La page change sur « Simulation ». | **15:23** | Le compte à été créer. La page de simulation c’est ouvert. | E.K. |
| 2 | Créer un compte | Créer un compte qui existe déjà. | Le compte ne sera pas créé.  Afficher une erreur. | **15:23** | Une erreur c’est afficher. | E.K. |
| 3 | Créer un compte | Créer un compte avec des champs vide. | Afficher une erreur. | **15:24** | Afficher une erreur – Les champs deviennent rouge | E.K. |
| 4 | Créer un compte | Créer un compte avec le champ « confirmation de mot de passe » qui est diffèrent du « mot de passe » | Afficher une erreur. | **15:24** | Afficher une erreur – Le champ deviens rouge | E.K. |
| 5 | Se logger | Se logger à un compte qui n’existe pas | Afficher une erreur. | **15:25** | Une erreur c’est afficher. | E.K. |
| 6 | Se logger | Se logger à un compte qui existe avec le mauvais mot de passe | Afficher une erreur. | **15:25** | Une erreur c’est afficher. | E.K. |
| 7 | Se logger | Se logger à un compte qui existe avec le bon mot de passe | La connexion va fonctionner.  La page change sur « Simulation ». | **15:26** | La connexion fonctionne. La page de simulation s’ouvre. | E.K. |
| 8 | Charger un jeu de réglages | Charger un jeu de réglages sans sélectionner un jeu | Afficher une erreur. | **15:26** | Afficher une erreur. | E.K. |
| 9 | Charger un jeu de réglages | Charger un jeu de réglages | Charger le jeu et afficher le succès | **15:27** | Un message de succès s’affiche et les réglages changent | E.K. |
| 10 | Sauvegarder un jeu de réglages | Sauvegarder un jeu de réglages sans sélectionner un jeu | Afficher une erreur. | **15:28** | Afficher une erreur. | E.K. |
| 11 | Sauvegarder un jeu de réglages | Sauvegarder un jeu de réglages | Sauvegarder dans la base donnée et afficher le succès | **15:28** | Un message de succès s’affiche. | E.K. |
| 12 | Créer un jeu de réglages | Créer un jeu de réglages sans nom | Afficher une erreur. | **15:29** | Afficher une erreur – Le champ deviens rouge | E.K. |
| 13 | Créer un jeu de réglages | Créer un jeu de réglages avec un nom qui existe déjà | Afficher une erreur. | **15:29** | Afficher une erreur. | E.K. |
| 14 | Créer un jeu de réglages | Créer un jeu de réglages avec un nom qui existe déjà mais qui a été ajouté après le lancement de l’application. Le jeu est donc dans la base de données mais pas la liste. | Afficher une erreur. | **15:29** | Afficher une erreur. | E.K. |
| 15 | Créer un jeu de réglages | Créer un nouveau jeu de réglages | Créer le jeu de réglages et afficher le succès | **15:30** | Un message de succès s’affiche et le jeu de réglage apparait dans la liste. | E.K. |
| 16 | Simulation | Lancer la simulation avec les réglages de départ | Le menu d’édition se ferme et la simulation fonctionne. | **15:30** | Le menu d’édition c’est bien fermé et la simulation tourne sans problème | E.K. |
| 17 | Simulation | Quitter la simulation. | La simulation s’arrête et le menu d’édition s’ouvre. | **15:30** | La simulation s’arrête et le menu s’ouvre. | E.K. |
| 18 | Simulation | Une fois que la simulation c’est stabilitée, vérifiez les statistiques sur la simulation. | La densité de la route équivaut au « In » et est le | **-** | Pas implémenter, pas testable | E.K. |
| 19 | Réglage général  Pixels par mètre | Changer le « Pixels par mètre » à une valeur inférieur à 0. | Afficher une erreur. | **15:31** | La simulation subie un effet miroir assez amusant. Vu que ceci ne cause aucun problème à l’utilisateur final et que je ne suis pas en avance, je laisserai la simulation dans cette état | E.K. |
| 20 | Réglage général  Pixels par mètre | Changer le « Pixels par mètre » à une valeur entre 1 et 100. | Change la taille et fonctionne normalement. | **15:33** | Change la taille et fonctionne normalement. | E.K. |
| 21 | Réglage général  Facteur de vitesse | Modifier le « Facteur de vitesse » à une valeur inférieur à 0. | Afficher une erreur. | **15:34** | Affiche une erreur et reset la valeur à 1. | E.K. |
| 22 | Réglage général  Facteur de vitesse | Modifier le « Facteur de vitesse » à une valeur entre 0.1 et 10. Une valeur plus grande pourrait causer un problème dans tous les cas. | Changer le facteur de vitesse. | **15:35** | La vitesse change mais pas le résultat de la simulation. | E.K. |
| 23 | Réglage véhicule  Décélération | Modifier cette valeur. | La valeur cause un changement lors des décélérations standard. | **15:35** | La valeur est prise en compte | E.K. |
| 24 | Réglage véhicule  Décélération % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 25 | Réglage véhicule  Décélération % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 26 | Réglage véhicule  Décélération % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 27 | Réglage véhicule  Décélération d’urgence | Modifier cette valeur. | La valeur cause un changement lors des décélérations d’urgence. | **15:36** | La valeur change mais elle n’est pas prise en compte vu qu’il n’y a pas de gestion des distances. | E.K. |
| 28 | Réglage véhicule  Décélération  d’urgence % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 29 | Réglage véhicule  Décélération  d’urgence % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 30 | Réglage véhicule  Décélération  d’urgence % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 31 | Réglage véhicule  Accélération | Modifier cette valeur. | La valeur cause un changement lors des décélérations d’urgence. | **15:37** | La valeur est prise en compte | E.K. |
| 32 | Réglage véhicule  Accélération % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 33 | Réglage véhicule  Accélération % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 34 | Réglage véhicule  Accélération % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 35 | Réglage véhicule  Temps de sécurité | Modifier cette valeur. | La valeur cause un changement de distance entre les véhicules | **-** | La valeur change mais elle n’est pas prise en compte vu qu’il n’y a pas de gestion des distances. | E.K. |
| 36 | Réglage véhicule  Temps de sécurité % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 37 | Réglage véhicule  Temps de sécurité % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 38 | Réglage véhicule  Temps de sécurité % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 39 | Réglage véhicule  Temps de réaction | Modifier cette valeur. | La valeur cause un changement de temps de réaction | **15:41** | La valeur change mais elle n’est pas prise en compte vu qu’il n’y a pas de gestion des temps de réaction. | E.K. |
| 40 | Réglage véhicule  Temps de réaction % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 41 | Réglage véhicule  Temps de réaction % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 42 | Réglage véhicule  Temps de réaction % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 43 | Cliquer sur une route | Nous cliquons sur un tronçon de route standard | Le menu de la route s’ouvre. | **15:43** | Le menu s’ouvre | E.K. |
| 44 | Cliquer sur un carrefour | Nous cliquons sur un carrefour. | Le menu de carrefour et le menu de feux de signalisation d’ouvre | **-** | Pas implémenter, pas testable | E.K. |
| 45 | Réglage route  Limitation de vitesse | Modifier cette valeur. | La valeur cause un changement de comportement des véhicules. | **15:44** | La valeur change bel et bien le comportement des véhicules | E.K. |
| 46 | Réglage route  Limitation de vitesse % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 47 | Réglage route  Limitation de vitesse % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 48 | Réglage route  Limitation de vitesse % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 49 | Réglage route  Taille | Modifier cette valeur. | La valeur cause un changement visuel ainsi qu’un changement de comportement des véhicules. | **15:45** | La taille visuelle ne change pas mais le comportement des véhicule change ! | E.K. |
| 50 | Réglage route  Densité | Modifier cette valeur. | La valeur cause un changement de la vitesse d’apparition des véhicules | **15:47** | Le nombre de véhicule par seconde change bel et bien ! | E.K. |
| 51 | Réglage route  Densité % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 52 | Réglage route  Densité % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 53 | Réglage route  Densité % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 54 | Réglage carrefour  Limitation de  vitesse | Modifier cette valeur. | La valeur cause un changement de comportement des véhicules. | **-** | Pas implémenter, pas testable | E.K. |
| 55 | Réglage carrefour  Limitation de  vitesse % | Modifie cette valeur avec une valeur inférieur à 0. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 56 | Réglage carrefour  Limitation de  vitesse % | Modifie cette valeur avec une valeur supérieur à 100. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 57 | Réglage carrefour  Limitation de  vitesse % | Modifie cette valeur avec une valeur entre 0 et 100. | Les véhicules subissent des différences de comportement plus élevé plus la valeur est grande. | **-** | Pas implémenter, pas testable | E.K. |
| 58 | Réglages feux  Temps vert | Modifie cette valeur avec une valeur inférieur à 0. À tester sur les 4 différents feux. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 59 | Réglages feux  Temps vert | Modifie cette valeur avec une valeur différente de 0. À tester sur les 4 différents feux. | La valeur cause un changement de temps de vert à feux tester. | **-** | Pas implémenter, pas testable | E.K. |
| 60 | Réglages feux  Temps orange | Modifie cette valeur avec une valeur inférieur à 0. À tester sur les 4 différents feux. | Afficher une erreur. | **-** | Pas implémenter, pas testable | E.K. |
| 61 | Réglages feux  Temps orange | Modifie cette valeur avec une valeur différente de 0. À tester sur les 4 différents feux. | A valeur cause un changement de temps d’orange à feux tester. | **-** | Pas implémenter, pas testable | E.K. |
| 62 | Réglages feux  Passage au rouge | Modifie les « croix ». À tester pour toutes les lignes et tous les feux. | Lors d’un passage au rouge, les lignes sélectionnées passent au vert | **-** | Pas implémenter, pas testable | E.K. |
| 63 | Réglages feux  Prochain | Appuyer sur le bouton « Prochain » | Un changement de feu en cours s’opère dans une direction constante (horaire ou anti-horaire, pas de hasard !) | - | Pas implémenter, pas testable | E.K. |

## Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
| Le 5 juin 2024 | Kuci Elvin | A close-up of a signature  Description automatically generated |

# Conclusion

Conclusion sur le projet en général. Les objectifs sont-ils atteints ? Si non, pourquoi ?

## Améliorations possibles

Y a-t-il des améliorations possibles (nouvelles fonctionnalités, amélioration et/ou modification du design, modification du support de données…) ?

## Auto-évaluation

Quels sont mes sentiments vis-à-vis du travail effectué. Enoncé des facteurs qui expliquent le succès ou l’échec de la réalisation des objectifs du projet.

# Bibliographie: liste des sources et références

Conseil: Cette partie peut contenir une liste des sources et éléments littéraires utilisés.

Aussi notions en modules etc

# Glossaire trier par ordre alphabetique

|  |  |
| --- | --- |
| Terme | Signification |
|  |  |
|  |  |
|  |  |

# Signatures

Je soussigné déclare que les informations contenues dans ce rapport de travail pratique individuel rendu ce jour le 06.06.2024 dans le cadre de la procédure de qualification de mon CFC d’informaticien, ne sont pas plagiées. Toutes les informations de sources extérieures ainsi que les informations fournies par des tiers durant le déroulement du travail sont consignées.

L’apprenti/-e doit signer la documentation avant la remise afin de témoigner de la justesse des informations qui y figurent.

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
| Le 6 juin 2024 | Elvin Kuci | A close-up of a signature  Description automatically generated |

# Annexes

Code imprimé, protocole de séances, etc.: spécifier complètement et annexer à la documentation.

Listings de scripts et programmes. Le travail fourni par la candidate ou le candidat doit être entièrement documenté. Omettre le code généré automatiquement s’il n’est pas nécessaire à la compréhension.

Dans le cas où des manuels ont été créés, vous pouvez les ajouter ici en tant qu’annexe.