REMERCIEMENT

17 janvier 2020

BORIBOUN Steven

ProxyDej

3 Rue carre de vigne, 44690, Monnières

# Table des matières

[II. Introduction 4](#_Toc32571429)

[1. Contexte 4](#_Toc32571430)

[1. La chaîne du froid 5](#_Toc32571431)

[2. La rupture de la chaîne du froid 5](#_Toc32571432)

[3. Les conséquences de cette rupture 5](#_Toc32571433)

[4. Le tableau de température 5](#_Toc32571434)

[2. Objectifs 6](#_Toc32571435)

[3. L’équipe 6](#_Toc32571436)

[4. Planification 6](#_Toc32571437)

[III. Spécifications Techniques 7](#_Toc32571438)

[1. Micro-ordinateurs 7](#_Toc32571439)

[2. Sonde de Température 8](#_Toc32571440)

[3. GPS 9](#_Toc32571441)

[4. Afficheur 10](#_Toc32571442)

[5. Langage De Programmation 11](#_Toc32571443)

[6. Alimentation de la Raspberry de livraison. 11](#_Toc32571444)

[7. Symfony 12](#_Toc32571445)

[IV. L’analyse 13](#_Toc32571446)

[1. Le diagramme de cas d’utilisation 13](#_Toc32571447)

[2. Les diagrammes de séquences 13](#_Toc32571448)

[3. La base de données sur MariaDB 15](#_Toc32571449)

[4. Interface WEB 15](#_Toc32571450)

[5. Partie Physique 15](#_Toc32571451)

[6. Bilan 16](#_Toc32571452)

# Introduction

## Contexte

Dans notre deuxième année de BTS Systèmes Numériques option Informatique et Réseaux, l’épreuve E6.2 est un projet technique. Il a pour objectif de nous placer en situation :

* De résolution d’un problème technique, tout en respectant une démarche conforme aux pratiques des entreprises :
  + Démarche de projet
  + Travailler en équipe
  + Respecter les contraintes, les normes, l’environnement etc…
* Acquérir de nouvelle connaissances, scientifiques, programmatiques, techniques et méthodologiques

A la fin de notre projet, nous effectuerons une soutenance d’une heure avec un jury composé de trois spécialistes (deux professeurs spécialisés et un professeur de physique appliqué). Cette soutenance sera composée de trois phases égale. Une phase de présentation général du projet. Une phase de mise en œuvre de projet et une phase d’interrogation exposé par le jury.

Dans ce dossier, vous trouverez un rendu représentatif de notre projet. Ce dossier comprend une partie commune, puis une partie allouée à chaque étudiant.

Dans notre cas, notre commanditaire est Proxidej. Leur besoin est de relever la température dans une camionnette frigorifique de livraison et dans une chambre froide. Leur but est de pouvoir justifier le respect de la chaîne du froid lors du transport de leurs plats préparés.

### La chaîne du froid

La chaîne du froid sont les opérations logistiques et domestiques (transport, manutention, stockage) visant à maintenir des produits à une température donnée afin de limiter la croissance bactérienne et de conserver les qualités gustatives. Tous les professionnels du secteur alimentaire se doivent de respecter la législation concernant les températures de conservation afin de maintenir la chaîne du froid aux niveaux de la fabrication, du stockage, du transport et de la distribution des produits. L'intérêt du terme *chaîne* est de souligner l'importance de la continuité des étapes, aucun maillon ne devant céder et anéantir l'essentiel de l'effort général déployé pour aboutir en fin de chaîne à un produit préservé de tout échauffement.

### La rupture de la chaîne du froid

Si celle-ci est brisée, les aliments dégèlent puis regèlent ce qui permet aux bactéries de se développer. Ce développement de ces bactéries, réduit considérablement la durée de vie du produit. La date indiquée sur l’étiquette, ne reste valable que si la chaîne du froid a été respecté et que nous stockons le produit dans la plage de température indiqué.

### Les conséquences de cette rupture

L'ingestion de ces bactéries peut alors entraîner une intoxication alimentaire dont les effets varient selon la fragilité de l'individu, pouvant être de la diarrhée, de la fièvre, ou aller jusqu'à la mort pour les cas les plus extrêmes. Plusieurs textes de lois encadrent l'hygiène alimentaire en restauration collective, et notamment la question de la chaîne du froid :

* Arrêté du 29 septembre 1997 : fixe les conditions d'hygiène applicables dans les établissements de restauration collective à caractère social.
* Arrêté du 20 juillet 1998 : fixe les conditions techniques et hygiéniques dans le transport des aliments.

### Le tableau de température

Chaque professionnel de l’industrie alimentaire est tenu de connaître et respecter scrupuleusement les températures de conservation réglementaires des denrées qu’il fabrique, stocke, transporte ou distribue. Voici donc ci-dessous, les températures à respecter selon les denrées.

|  |  |
| --- | --- |
| Catégories de denrées | Température de conservation |
| Poissons, crustacés et mollusques cuits, viandes cuites | Entre 0° C à + 4° C |
| Viandes crues, poissons non cuits, charcuteries, etc… | + 4° C maximum |
| Fruits et légumes prêts à l'emploi | + 4° C |

## Objectifs

Les objectifs de ce projet permettront de mettre en œuvre plusieurs dispositifs de contrôle de température. Ceci afin d’assurer la bonne conservation des repas livrés. Ces systèmes pourront ainsi détecter les incidents, mais aussi de prouver le respect de la chaîne de froid en cas de contrôle sanitaire.

## L’équipe

Notre équipe est composée de trois étudiants de BTS SN-IR en deuxième année :

* Florian ENET
* Steven BORIBOUN
* Matthew FLENET

Nous avons réparti nos tâches, selon nos diverses expériences professionnelles et personnelles. Afin de faire découvrir ou de réaliser les parties souhaitées.

## Planification

La partie planification est importante pour le bon déroulement d’un projet. Pour ainsi répondre au cahier des charges, les objectifs et les délais, nous allons utiliser un diagramme de Gantt. Nous permettant ainsi de savoir, les tâches à réaliser et quand les réaliser. Vous trouverez notre diagramme de Gantt dans l’annexe.

M.ENET sera responsable du développement de l’application Web de surveillance des températures et de suivi des tournées.

M.FLENET sera responsable du développement du système embarqué de la chambre froide.

M.BORIBOUN sera responsable du développement du système embarqué d’un véhicule.

# Spécifications Techniques

## Micro-ordinateurs

Dans le contexte du projet, il nous faut deux **micro-ordinateurs**, pour la gestion des deux capteurs de température et du capteur de géolocalisation. Ces micro-ordinateurs permettront de gérer le flux d’information reçu et l’enverra sur le serveur de l’entreprise.

Les Micro-ordinateurs devront respecter ces conditions :

* Doit posséder un **espace de stockage**.
* Doit pouvoir **se connecter en Wi-Fi**.
* Doit pouvoir **résister un minimum le froid**.
* Doit avoir des **pins des entrées et des sorties**.

Les deux propositions de micro-ordinateurs sont :

* **Raspberry PI3B+**
* **Arduino UNO Rev3**

Ces deux **micro-ordinateurs** sont connus dans leur milieu. Néanmoins, ils ne sont **pas faits pour les mêmes utilisations**, nous listerons leurs caractéristiques afin de choisir le plus **adapté**.

Vous trouverez ci-dessous le tableau comparatif entre le Raspberry PI3B+ et l’Arduino UNO Rev3 :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Raspberry PI3B+** | **Arduino UNO Rev3** | **Meilleur choix** |
| Consommation électrique | 5V | 7-12V | Raspberry |
| Température de fonctionnement | 0-50°C | Non communiqué | Raspberry |
| Ports entrées/sorties | 40 pins | 12 pins | Raspberry |
| Langage de programmation | Python | C | Raspberry |
| Mémoire vive | 1GB | 2 kB | Raspberry |
| Emplacement carte SD | Oui | Non | Raspberry |
| Connexion Wi-Fi | Oui | Non | Raspberry |
| Prix | ≃ 40€ | ≃ 20€ | Arduino |

Notre choix final sera donc, le **Raspberry PI3B+**. Il nous faudra quatre Raspberry, un pour chaque véhicule de livraison, plus undans la chambre froide.

Pour le fonctionnement d’une carte Raspberry, il faut :

* Un **Système d’alimentation 5V** environ
* Une **carte micro SD** avec un **système d’exploitation flashé**

**Pour la simplification du projet, nous allons installer et programmer une seule carte (celles qui seront dans les véhicules). Ensuite, nous allons la cloner pour les deux autres véhicules.**

## Sonde de Température

Nous avons besoin d’utiliser des **Sondes de température** afin de surveiller la Température d’une chambre froide ainsi qu'à l’intérieur des véhicules de livraison dans le but d’effectuer une traçabilité du froid des véhicules et de la chambre.

Les Sondes Que nous aurons à choisir devront respecter ces conditions :

* Doit pouvoir être **connectée à un Raspberry** directement ou **via GPIO.**
* Doit être la **moins encombrante** **possible** (plus petite, moins de câble).
* Doit être **peu onéreuse**.
* Doit pouvoir capter une plage de température comprise entre **0-5°C**

Les deux Sonde à nos dispositions sont les suivantes :

* **DS18B20**
* **DB130-10**

La **DS18B20** est une sonde assez connue dans le milieu des sondes de températures et est produite par l’entreprise **Maxim integrated**. En opposition peu d’informations sont à disposition sur la **DB130-10**, nous utiliserons les caractéristiques de celles produites par l’entreprise **DFROBOT** pour nos comparatifs.

Veuillez trouver ci-dessous un comparatif des caractéristiques de chacune des sondes ainsi que celle que nous avons choisi pour ce projet :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **DS18B20** | **DB130-10** | **Meilleur Choix** |
| **Consommation électrique** | 3V à 5,5V | 5V | **DB130-10** |
| **Type d’interface** | Câble unique | Interface 2 câbles | **DB130-10** |
| **Plage de température** | -55°C à +125°C | -10°C à 80°C | **Indifférent** |
| **Taille** | 30mm x 6mm | 49mm x 14mm | **DS18B20** |
| **Longueur du Cable** | 91cm | 90 cm | **DB130-10** |
| **Prix** | 10€ | 27€ (29.50$) | **DS18B20** |

Notre Choix de Sonde final sera donc la **DB130-10**, il nous en faudra 4, une pour la chambre froide et une pour chaque Véhicule de Livraison.

Nous avons trouvé plus intéressant de **conserver la même sonde pour chaque cas d’utilisation.**

## GPS

Nous serons amenées à récupérer des **informations GPS** pour reproduire le parcours d’un véhicule de livraison et récupérer toutes les informations souhaitées en fonction d’une position a un temps donné. Par conséquent des r**écepteurs GPS** sera nécessaire.

Les Récepteurs GPS Que nous aurons à choisir devront respecter ces conditions :

* Doit avoir une **faible consommation électrique.**
* Doit être la **moins encombrante possible** (plus petite).
* Doit être **peu onéreuse**.

Les deux Récepteurs GPS à nos dispositions sont les suivants :

* **BU-353**
* **Here 2**

Le récepteur GPS BU-353 est produit par **Globalsat**, une entreprise spécialisée dans les **GPS Tracker, récepteurs GPS...**

**Pixhawk 2** est un **projet** datant de **2008** qui vise à fournir du **matériel de contrôle de pilotage** (autopilote) **haut de gamme**, à **coûts bas ou raisonnables** à des communautés d'utilisateurs universitaires, de loisirs et industrielles.

Veuillez trouver ci-dessous un comparatif des caractéristiques de chacun des Récepteurs GPS ainsi que celui que nous avons choisi pour ce projet :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **BU-353** | **SIM28** | **Meilleur Choix** |
| **Chipset GPS** | L1 SiRF Star III | UB-5010 | **SIM28** |
| **Interface** | USB | GPIO | **SIM28** |
| **Fréquence trames NMEA** | 1 Hz | Max 10 Hz | **SIM28** |
| **Récepteur 20 canaux** | L1, C/A code | L1, C/A, WAAS, EGNOS, MSAS, GAGAN | **SIM28** |
| **Dimensions** | 53mm\*19mm | 53mm\*47mm\*23mm | **SIM28** |
| **Consommation** | 80 mA | 35 mA | **SIM28** |
| **Précision** | 10m (2D RMS) | 5 m | **SIM28** |
| **Prix** | 39€ | 23€ | **SIM28** |

On choisira par conséquent Le **SIM28** car il est plus **économique** et **plus performant**.

## Afficheur

Pour pouvoir afficher les données de la sonde chargée d’observer la température, nous aurons besoin d’un **afficheur** à l’entrée de la chambre froide nous donnant toutes les informations que l’on souhaite

Les différents **afficheurs** que nous aurons à choisir devront respecter ces conditions :

* Doit avoir une **faible consommation électrique.**
* Doit être la **moins encombrante possible** (plus petite).
* Doit être **peu onéreuse**.
* Doit être **tactile.**

Les deux afficheurs à nos dispositions sont les suivants :

* **Raspberry Pi 7’’ Touchscreen Display**
* **Raspad**

Le récepteur GPS BU-353 est produit par **Globalsat**, une entreprise spécialisée dans les **GPS Tracker, récepteurs Gps...**

**Pixhawk 2** est un **projet** datant de **2008** qui vise à fournir du **matériel de contrôle de pilotage** (autopilote) **haut de gamme**, à **coûts bas ou raisonnables** à des communautés d'utilisateurs universitaires, de loisirs et industrielles.

Veuillez trouver ci-dessous un comparatif des caractéristiques de chacun des afficheurs ainsi que celui que nous avons choisi pour ce projet :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Touchscreen Display** | **Raspad** | **Meilleur Choix** |
| **Consommation** | Avec la Raspberry (3 à 5,5V) | 6000mAh | **Touchscreen Display** |
| **Dimension** | 194mm x 110mm x 20mm | 25,4 cm de diagonale | **Touchscreen Display** |
| **Tactile** | Oui | Oui | **Equivalent** |
| **Connectique** | DSI port | USB | **Touchscreen Display** |
| **Résolution** | 800\*480 pixels | 1280\*800 pixels | **Raspad** |
| **Fréquence** | 60Hz | 60Hz | **Equivalent** |
| **Prix** | 60€ | 200€ | **Touchscreen Display** |

On choisira par conséquent Le **Raspberry Pi 7’’ Touchscreen Display** car il est plus économique et **moins** **consommateur** en énergie. De plus sa **connectique** est plus adaptée dans notre contexte.

## Langage De Programmation

Enfin la partie la plus importante à nos yeux, nous devons choisir un langage de programmation dans lequel

Les différents **afficheurs** que nous aurons à choisir devront respecter ces conditions :

* Doit être **facile** **d’utilisation**
* Doit être **dynamique (**pour le **temps réel)**

Les deux langages à notre disposition sont :

* **Python**
* **C++**

D’autres langages peuvent être utilisés mais ces deux-là sont les deux **meilleures options** sur un système comme une **Raspberry**

Veuillez trouver ci-dessous un comparatif des caractéristiques de chacun des afficheurs ainsi que celui que nous avons choisi pour ce projet :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Python** | **C++** | **Meilleur Choix** |
| **Compilation** | Besoin d’un interpréteur | Pré-compilé avant exécution | **C++** |
| **Efficience** | Language Simplifié | Language traditionnel | **Python** |
| **Difficulté** | Facile | Difficile | **Python** |
| **Collection des déchets** | Oui | Non | **Python** |
| **Cross-platform** | Oui | Non | **Python** |
| **Langage Dynamique** | Oui | Non | **Python** |

On choisira par conséquent Le **Python**, car parmi les deux il est le seul langage dynamique, de plus il est beaucoup plus **simple** d’utilisation que le **C++.**

## Alimentation de la Raspberry de livraison.

La Raspberry a besoin d’alimentation de 5V 2.5A en micro USB pour fonctionner.

Il faut que l’alimentation soit :

* Peu consommatrice en énergie (5V 2.5A).
* Que l’alimentation soit éteinte quand la voiture est à l’arrêt

Nous avons deux solutions qui sont à notre disposition :

* Un adaptateur allume-cigare à micro USB de 5V
* Régulateur 5V/12V

Veuillez trouver ci-dessous un comparatif des caractéristiques de chacun des afficheurs ainsi que celui que nous avons choisi pour ce projet :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Adaptateur allume-cigare 5V** | **Régulateur 5V/12V** | **Meilleur Choix** |
| **Consommation** | Sortie : 5V/1A | Sortie : 5V/2.4A | **Régulateur 5V/12V** |
| **Longueur des câbles** | 1.20 m | 5 cm | **Régulateur 5V/12V** |
| **Difficulté de mise en place** | Plug-and-Play | Besoin d’une installation | **Adaptateur allume-cigare 5V** |
| **Courant qui passe plus quand la voiture et à l’arrêt** | Oui | Oui | **Equivalent** |
| **Protection de surintensité** | Non | Oui | **Régulateur 5V/12V** |
| **Prix** | 10.90€ | 13.90€ | **Adaptateur allume-cigare 5V** |

On choisira par conséquent **Régulateur 5V/12V**, car parmi les deux, c’est celui qui gardera un courant stable peu importe l’intensité de la voiture.

## NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML.

## Symfony

Symfony est un framework français open source pour PHP. Les frameworks, en général, permettent un gain de productivité mais pas seulement. Ils permettent d'avoir une bonne organisation du code et on sait qu'un code bien organisé est plus maintenable et évolutif. Symfony est open source. On n'a donc pas de contraintes imposées et on peut développer des solutions propriétaires. Il est considéré comme un des framework PHP les plus puissants et les plus flexibles. Un des principaux inconvénients d'un framework est le temps d'apprentissage qui est bien plus élevé que pour un langage en lui-même. Symfony, plus particulièrement nécessite d'apprendre plusieurs autres technologies qui lui sont propres comme YAML ou Doctrine.

Une Doctrine ORM implémente 2 patterns objets pour mapper un objet PHP à des éléments d'un système de persistance :

* Le pattern "Data Mapper".
* Le pattern "Unit of Work".

Le Data Mapper est une couche qui synchronise la donnée stockée en base avec les objets PHP. En d'autres termes :

* Il peut insérer, mettre à jour des entrées en base de données à partir de données contenues dans les propriétés d'un objet ;
* Il peut supprimer des entrées en base de données si les "entités" liées sont identifiées pour être supprimées ;
* Il "hydrate" des objets en mémoire à partir d'informations contenues en base.

L'implémentation dans le projet Doctrine de ce Data Mapper s'appelle l'Entity Manager, les entités ne sont que de simples objets PHP mappés.

L’utilisation de Symfony, nous permettra de créer un serveur internet virtuel, pour notre phase de conception.

# L’analyse

## Le diagramme de cas d’utilisation

Le diagramme de cas d’utilisation permet de montrer les différents types de scenarios que les acteurs peuvent rencontrer lors de l’utilisation d’un projet. Ces scenarios permettront de créer des diagrammes de séquences. Voici ci-dessous le diagramme de cas d’utilisations pour notre projet :



Figure Diagramme de cas d'utilisation du projet

## Les diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Voici le premier diagramme Acquisition des données Température et GPS de la Raspberry de livraison :

Une image contenant texte, carte

Description générée automatiquement

Figure Diagramme de séquence pour l’acquisition

Voici le deuxième diagramme Synchronisation des données Température et GPS de la Raspberry de livraison :

Une image contenant texte, carte

Description générée automatiquement

Figure Diagramme de séquence pour la synchronisation

Voici le diagramme orienté pour le site WEB, ce diagramme ci-dessous permet de consulter les données.

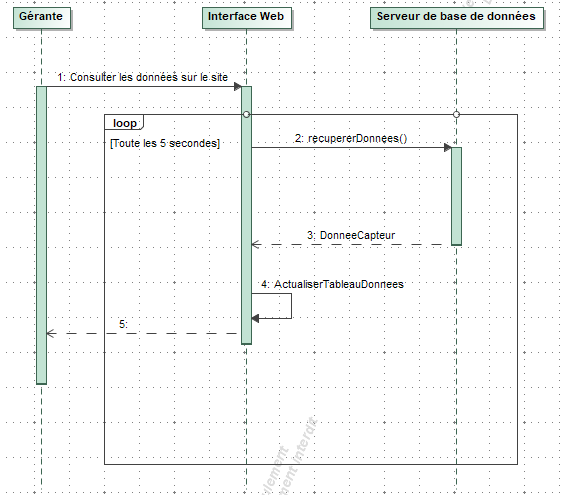


Figure Diagramme de séquence pour la consultation des données

Le diagramme ci-dessous, nous montre comment nous récupérons les informations de température de la chambre froide.

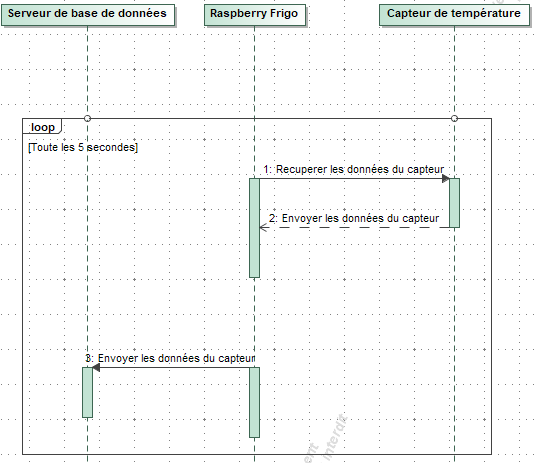


Figure Diagramme de séquence pour la chambre froide

Le diagramme ci-dessous, nous montre comment nous récupérons les informations de température d’un véhicule frigorifique.

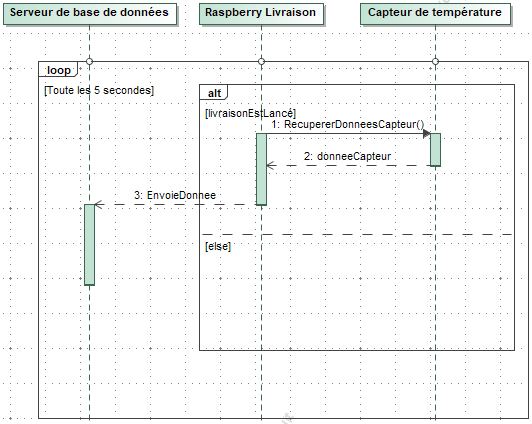


Figure Diagramme de séquence pour un véhicule livraison

Notre dernier diagramme permettra d’alerter la gérante en cas de problèmes de température.

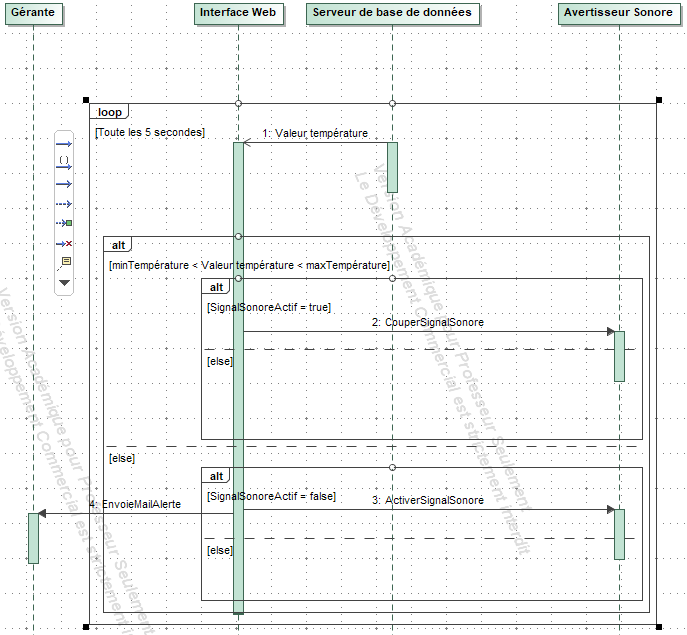


Figure Diagramme de séquence pour alerter la gérante

## Diagramme de classe

Le diagramme de classe permet de présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Le diagramme ci-dessous représente la partie des systèmes de relevé de température et de données GPS.

Une image contenant carte, texte

Description générée automatiquement

Figure Diagramme de classe pour la Raspberry de livraison

Le diagramme ci-dessous, nous montre les classes à réaliser pour l’application WEB.

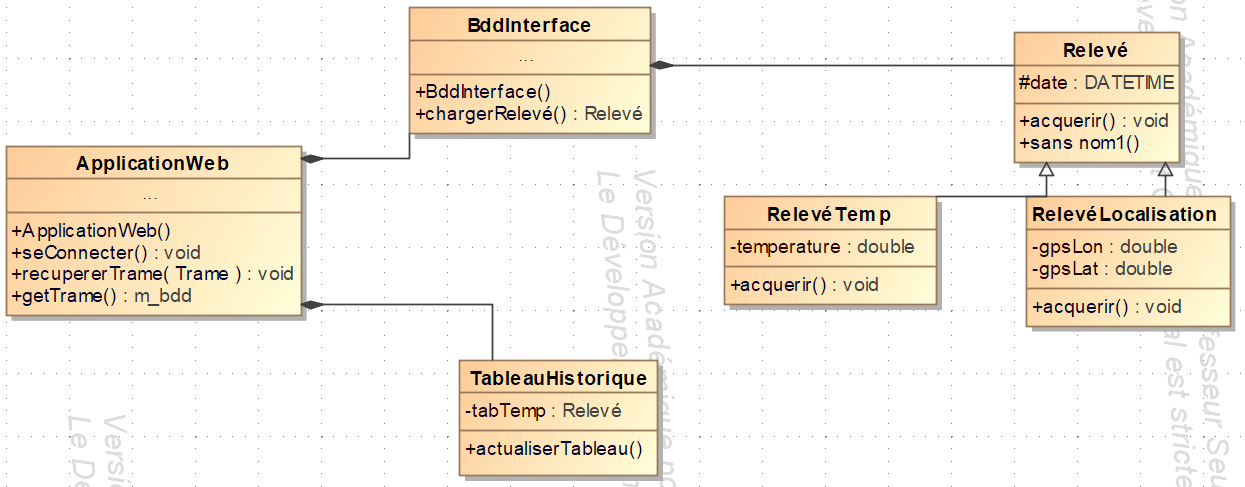


Figure Diagramme de classe pour l'application Web

## La base de données sur MariaDB

La base de données permettra de stocker les informations des différentes Raspberry. Elle possèdera trois colonnes, l’une se nommera « ChambreFroide », l’une « Livraison » et la dernière sera « users ». Les deux premières colonnes récupèreront les informations du capteur de température lié à sa Raspberry. La particularité du deuxième est qu’il récupèrera en plus les informations de longitude et de latitude du capteur GPS lié à sa Raspberry. La dernière colonne permettra à la gérante du site de pouvoir consulter des informations, classées « confidentiel ». Il y aura donc le statut « admin » et le statut « user ». Un mail devra être adresser à la gérante, donc il faudra qu’elle renseigne son adresse mail dans la colonne adéquate.



Figure Diagramme de la base de données

## Interface WEB

Notre interface WEB permettra à la gérante de pouvoir visualiser les informations stockées dans la base de données. La gérante devra cependant se connecter avec ses identifiants afin d’accéder à ces informations. Le site aura une page « Suivie livraison », « Chambre Froide » et « Historique ». C’est ce même site qui enverra un courriel à la gérante, si un incident concernant la conservation des aliments intervient. Vous trouverez les prototypes de ces pages en annexe.

## Charte graphique

Notre charte graphique sera composée des deux couleurs de l’entreprise. Le rouge pourpre (#a40233) et le jaune (#fbb021). Ces couleurs seront les couleurs de notre site et du graphique nous permettant de voir la température de la chambre froide.

## Partie Physique

### Changement de température (ENET)

L’une des parties physiques de notre projet se posera sur la thermodynamique du changement d’état de refroidissement. Le refroidissement est une perte de puissance colorifique. Elle peut se perdre de trois façons :

* La perte provenant du rayonnement du corps (exemple : fer rouge proche, nous chauffe par ses infra-rouges)
* La perte par conduction (exemple : la peau se chauffe si on la plonge dans de l’eau chaude)
* La perte par convection (exemple : un mouvement d'air chaud réchauffe la peau beaucoup mieux que de l’air stagnant à même température)

La façon qui nous intéresse est la troisième, la convection. Cette façon correspond à la manière avec laquelle, nos plats peuvent varier de température. Car pour être plus précis que l’exemple ci-dessus, voici sa définition. La convection se fait grâce au contact du fluide ambiant/avec le corps, ce qui arrache de petits éléments de matière plus important quand il y a fort brassage volumique du fluide.

Cette formule de perte s’écrit : **(Pc = h1.S.∆T)** où h1(W/m²-K) est le coefficient de transfert thermique, S (m²) est la surface et DT la différence de température.

### Matthew

### Steven

## Conclusion de fin de projet

# Annexes

## Diagramme de Gantt

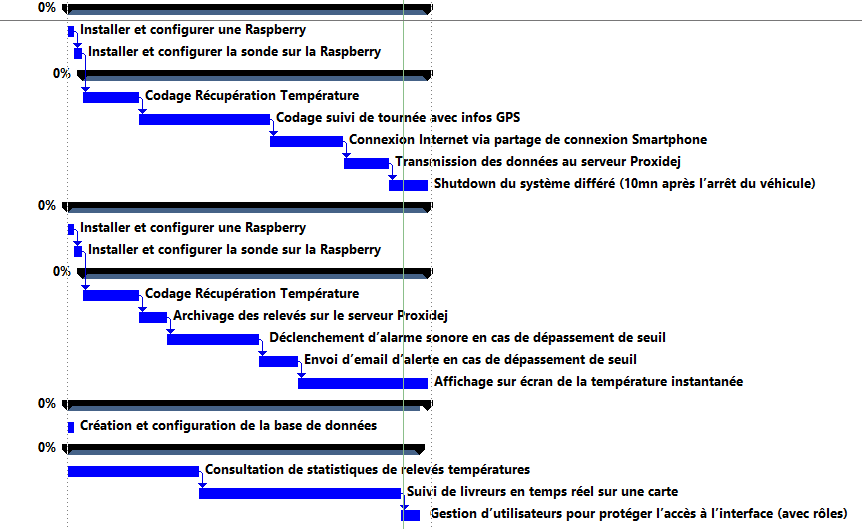


Figure Diagramme de Gantt du projet SFL2

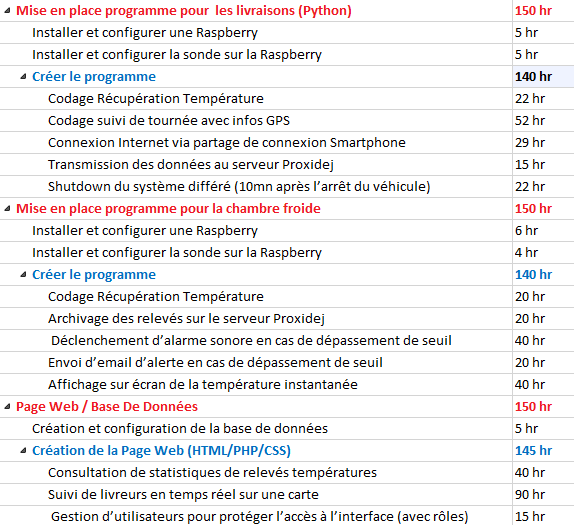


Figure Tableau des tâches du projet SFL2

## Prototypes de page WEB

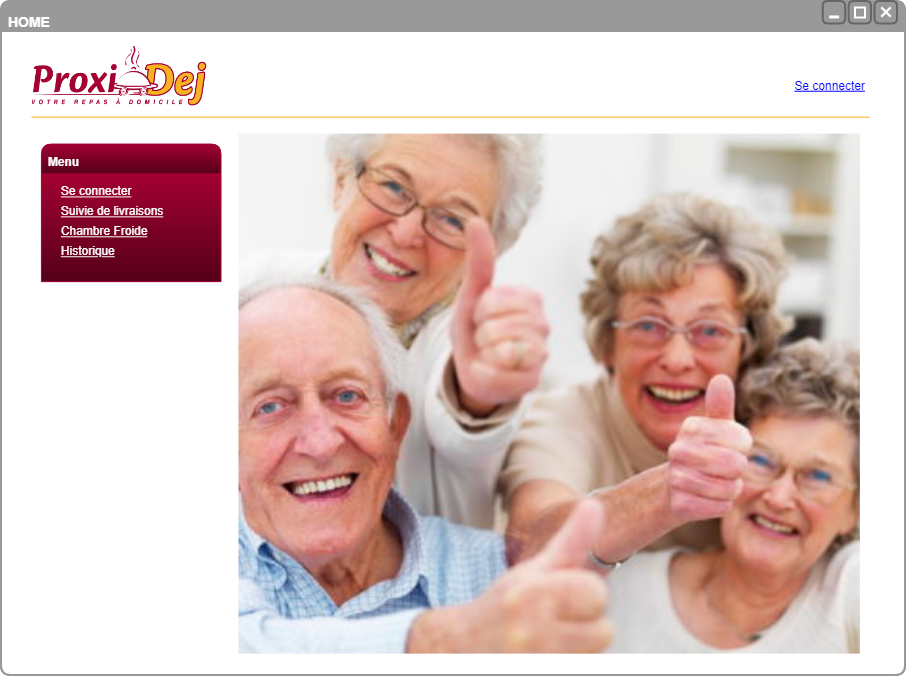


Figure Page d'acceuil du site Proxidej

La page de connexion sera une étape obligatoire pour accéder aux pages suivantes ci-dessous.

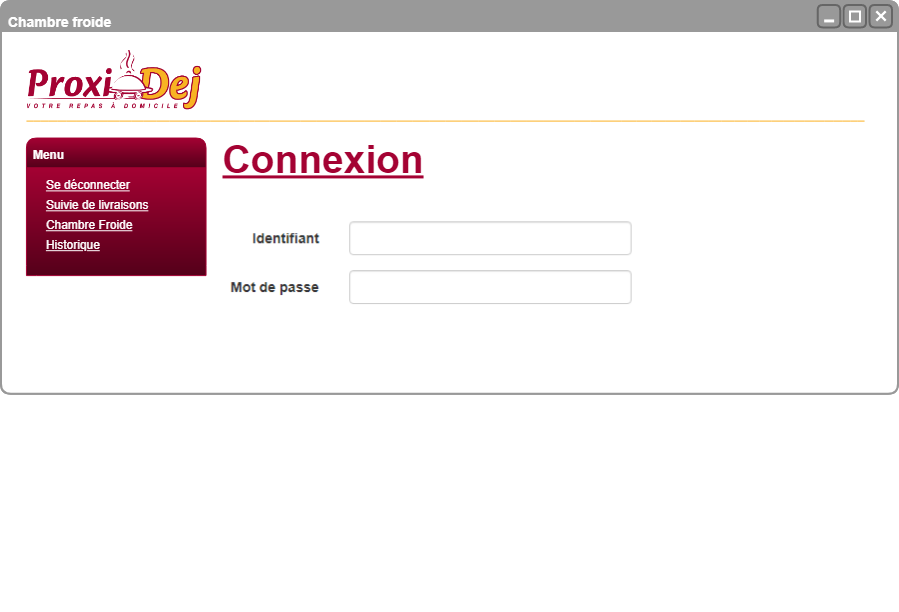


Figure Page de connexion

La page de livraison possède une map. Nous avons un overlay en haut à droite qui nous permet de choisir d’afficher ou non, les points et le trajet d’une livraison. Les trajets sont triés selon les jours.



Figure Page de suivi avec l'overlay

La page permettant la surveillance de la chambre froide, nous permet de trier les données selon la date. Dans notre graphique, nous possédons les valeurs de la journée par tranche de cinq minutes, soit 288 valeurs.

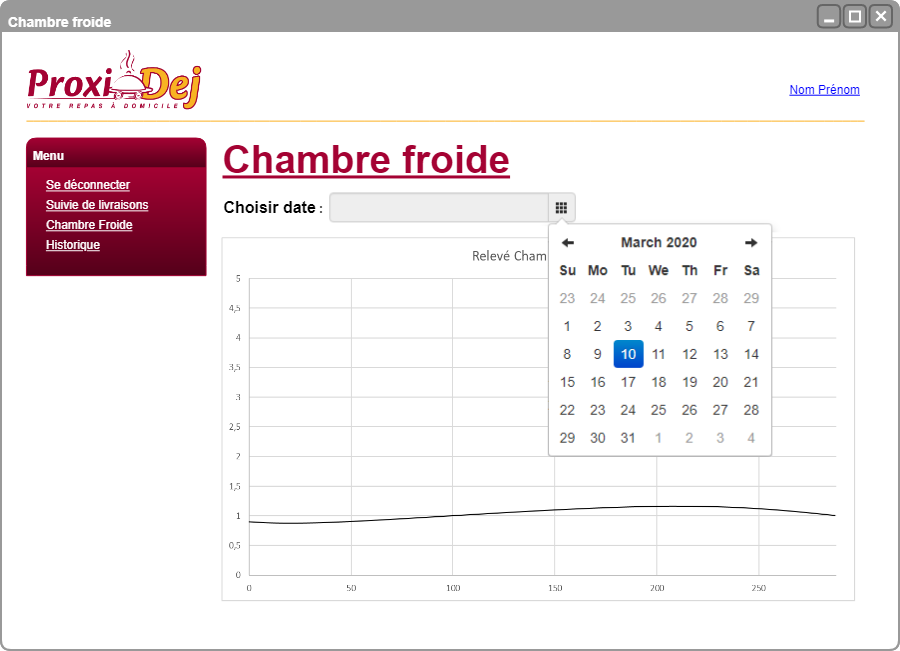


Figure Page de surveillance de la chambre froide

