

Spécifications et Analyse

**Serre Automatique**

Système d’acquisition

Ce document regroupe l’analyse du projet, ses spécifications ainsi que les outils liés à son développement.

Lucas Minaud, Audran Raynal, Killian Labattut, Valentin Chevallier

Créé le 08/01/2019



Table des matières

[I. Présentation du projet 2](#_Toc9509352)

[**1)** **Enoncé général du besoin** 2](#_Toc9509353)

[**2)** **Aspect contractuel du projet** 2](#_Toc9509354)

[II. Expression du besoin 2](#_Toc9509355)

[III. Identification des équipements 3](#_Toc9509356)

[**1)** **Synoptique du système** 3](#_Toc9509357)

[**2)** **Description des outils** 4](#_Toc9509358)

[IV. Identification du besoin 4](#_Toc9509359)

[**1)** **Besoins fonctionnels** 4](#_Toc9509360)

[**2)** **Besoins non-fonctionnels** 5](#_Toc9509361)

[V. Ressources mise à dispositions 5](#_Toc9509362)

[**1)** **Ressources matérielles** 5](#_Toc9509363)

[**2)** **Ressources logicielles** 6](#_Toc9509364)

[VI. Analyse 6](#_Toc9509365)

[**1)** **Tâches à réaliser** 6](#_Toc9509366)

[**2)** **Suivi de gestion** 7](#_Toc9509367)

[a) Analyse et choix de début de projet 7](#_Toc9509368)

[b) Choix définitifs et modifications éventuelles apportés 7](#_Toc9509369)

[**3)** **Identification des acteurs** 8](#_Toc9509370)

[VII. Intégration et versionning 9](#_Toc9509371)

[VIII. Diagramme 10](#_Toc9509372)

[**1)** **Diagramme d’exigences** 10](#_Toc9509373)

[**2)** **Diagrammes des cas d’utilisation** 11](#_Toc9509374)

[**3)** **Diagrammes de séquences** 12](#_Toc9509375)

**[4)](#_Toc9509376)****[Diagrammes de classes](#_Toc9509376)** [14](#_Toc9509376)

**[5)](#_Toc9509377)****[Diagramme de Gantt](#_Toc9509377)** [15](#_Toc9509377)

[**6)** **Diagramme de Base de données** 16](#_Toc9509378)

[**7)** **Diagramme de Déploiement** 17](#_Toc9509379)

# Présentation du projet

## **Enoncé général du besoin**

L’entreprise Groupe Olivier est le client, il est commanditaire du projet de serre automatique. En effet une serre augmente les possibilités de culture. Son équipement pourrait disposer d’appareils automatique pour assurer : l’arrosage, le chauffage, aération etc.

Grâce à la serre, le jardinier peut cultiver sans se soucier des intempéries que subissent les plantes dans leur environnement naturel. Il existe deux types de serres :

- les serres froides

- les serres chaudes

## **Aspect contractuel du projet**

|  |  |
| --- | --- |
| Commanditaire | Groupe Olivier |
| Acteurs | Jardinier |
| Temps de réalisation | Du 8 janvier au 30 mai |
| Equipe de développement | 4 étudiants |
| Professeur référent | M. Sébastien Angibaud |

# Expression du besoin

Le Groupe Olivier a besoin d’automatiser une serre pour effectuer les tâches les plus astreignantes. Pour se faire il se doit de détecter quand est-ce que la serre nécessite d’être entretenue.

De ce fait le client souhaite : récupérer les mesures nécessaires, posséder un système situé dans un local afin de mémoriser les données. Consulter les mesures en temps réel par des histogrammes. Être informé en temps réel de l’état de fonctionnement de l’ensemble du système.

L’application mobile servira à l’exploitant et au jardinier, ils pourront veiller sur l’environnement et les plants de la serre.

L’équipe de développement à 260 heures toutes comprises pour mener à bien ce projet, si le projet abouti l’exploitant de la ferme gagnera en bénéfices car il dispensera son ouvrier de certaines tâches pénibles.

# Identification des équipements

* Capteur de température intérieure SDTS ARIA
* Capteur de température d’eau chaude SDTS ARIA
* Pluviomètre Rain Collector II DAVIS
* Capteur d’humidité intérieure DHT22
* Capteur de lumière TSL2591
* Anémomètre-girouette 7911 DAVIS

## **Synoptique du système**



## **Description des outils**

* La serre est la pièce où sont effectuées les mesures à intervalles
* Le local proche de la serre est le lieu où se trouve le PC d’archivage
* La carte de gestion est la carte principale où toutes les données sont collectées et archivées.
* La carte d’interface est une carte d’entrées - sorties grâce à laquelle les capteurs sont reliés à la carte principale, par ses nombreuses entrées.
* Les capteurs : capteur d’hygrométrie, de pluviométrie, de température intérieur et d’eau, force et direction du vent.
* Le smartphone Android : il sert à visualiser l’état de fonctionnement de l’ensemble du système
* Le système de sauvegarde est une base de données, où sont enregistrées tous les relevés journaliers.
* L’application WEB : elle sert à visualiser l’évolution des mesures sur une période demandée pouvant s’étendre jusqu’à un an. Ainsi que l’application Android il sert à contrôler le fonctionnement du système.

# Identification du besoin

## **Besoins fonctionnels**

* Réaliser des relevés :

Les capteurs transmettent les données à la carte de gestion.

* Visualiser l’évolution d’une mesure :

La carte de gestion écrit les données dans la base de données. L’application WEB affiche les relevés enregistrés depuis la base de données, l’exploitant choisit une période.

* Visualiser l’état de la serre :

L’exploitant consultera l’application WEB pour afficher les dernières mesures enregistrées depuis la base de données.

* Visualiser l’état du système :

L’exploitant consulte les applications qui affichent l’état des capteurs. La carte de gestion collecte les informations d’état du système et les enregistrent sur la base de données.

* Définir la périodicité des mesures :

L’exploitant choisira selon son besoin.

## **Besoins non-fonctionnels**

**Développement**

L’application Android sera développée en Java, sous Android Studio.

Les programmes d’acquisition des données, ainsi que le programme de base de la carte de gestion seront développés en Python 3

Le langage UML sera utilisé pour permettre une analyse et une compréhension poussé du projet.

**Contraintes financières**

L’achat du capteur d’intensité lumineuse sera à prévoir. La section SN nous octroie un budget de 100 euros pour ce capteur et d’éventuel autres achats.

**Contraintes qualité**

L’exploitant devra pouvoir changer un capteur ou en rajouter un sans devoir redémarrer le système ou le modifier.

**Contraintes de fiabilité, sécurité**

# Ressources mise à dispositions

## **Ressources matérielles**

Les ressources matérielles sont fournies par Saint Félix-La Salle :

* Les PC sous Windows 10 au nombre de quatre.
* Un serveur Debian 8.7 hébergeant la base de données.
* Les capteurs suivants : anémomètre-girouette, hygromètre, capteur de température intérieure, capteur de température de l’eau et pluviomètre.
* Le capteur d’intensité lumineuse a été commandé 8 euros en ligne.
* Raspberry Pi 3.
* Arduino MEGA/UNO .
* Smartphone Android
* Maquette de la serre en kit
* 2 capteurs de température ARIA pt-100 mis en place avec un Shield Libellium 4-20 Ma

## **Ressources logicielles**

Pour la documentation, revues nous utilisons la suite Office : Word et Powerpoint

Pour le diagramme de Gantt : Microsoft Project 2013

Pour les divers diagrammes UML : MagicDraw UML avec le plugin SysML

Pour les diagrammes réseau : Edraw ou Microsoft Paint

GitHub

GitHub est une plateforme open source se basant sur un logiciel de gestion de version, permettant aux développeurs de logiciel de partager et de collaborer.

Les logiciels suivants nous sont proposés :

* Android Studio pour le développement de l’application Android
* Le Framework Symfony avec l’environnement NetBeans

# Analyse

## **Tâches à réaliser**

Etudiant n°1 :

Installation de la carte gestion

Développement du programme de base de la carte gestion

Etude, gestion du protocole de l’anémomètre-girouette

Acquisition de la mesure direction et force du vent

Etudiant n°2 :

Etude et choix des capteurs

Réalisation de l’application web de visualisation périodique

Acquisition de la mesure hygrométrie et de la luminosité

Etudiant n°3 :

Installation et configuration de la base de données

Réalisation de l’application web de visualisation en temps réel

Acquisition de la mesure température sous serre et température de l’eau

Etudiant n°4 :

Réalisation de l’application Android

Acquisition de la mesure pluviométrie

## **Suivi de gestion**

### Analyse et choix de début de projet

Etudiant n°1 :

Une version Raspbian est installée sur une carte Raspberry Pi. C’est la carte gestion.

Le programme de gestion est développé en Python.

Les capteurs anémomètre et girouette sont reliés à la carte de gestion, grâce à un adaptateur RJ-11 / USB.

Les programmes sont en Python.

Etudiant n°2 :

NetBeans : logiciel de programmation afin de pouvoir créer l’interface WEB.

Choix de l’outil Symfony pour développer l’interface web de façon structuré et claire.

Etudiant n°3 :

Base de données créer sur un serveur distant grâce à l’application « MySQL ».

Choix de l’outil Symfony pour développer l’interface web en passant par le serveur contenant la base de données.

Récupération des valeurs de température via une carte Arduino et un Shield 4-20 Ma relié à la carte.

Etudiant n°4 :

Réalisation de l’application Android

Acquisition de la mesure pluviométrie

### Choix définitifs et modifications éventuelles apportés

Etudiant n°1 :

La version Raspbian (Raspbian Jessie) est bien installée et communique.

Le programme est en développement.

Les capteurs anémomètre et girouette seront finalement relié directement au Port Analogique de la carte d’acquisition. (Modifié)

Les programmes sont en Python.

Etudiant n°2 :

NetBeans : logiciel de programmation afin de pouvoir créer l’interface WEB.

Capteur de luminosité : tsl2591.

Capteur d’hygrométrie : Temperature&Humidity Sensor.

Logiciel WinSCP afin d’avoir un accès simplifié pour le serveur.

Etudiant n°3 :

Base de données créer et fonctionnelle

Choix de Symfony abandonné pour rester sur un format html/PHP plus basique et plus clair

Capteur de températures mis en place et fonctionnel, le programme présent sur la carte d’acquisition récupérant les données est terminé.

Etudiant n°4 :

Le pluviomètre sera finalement relié à la carte d’acquisition par RJ11, et non pas par un protocole sans-fil 433Mhz.

L’API de développement de l’application sera la 22ème, car elle est très répandue parmi les usagers tout en étant assez récente.

## **Identification des acteurs**

L’exploitant de la serre :

Il est l’acteur principal du système, faisant partie de l’entreprise commanditaire. Il devra pouvoir visualiser grâce à l’application Android l’état de la serre et donc des capteurs, et de son état de fonctionnement en temps réel.

Il devra également pouvoir consulter les mesures stockées, en choisissant la période de la visualisation, c’est-à-dire jour/semaine/mois/année.

L’exploitant pourra utiliser le système sans compétences informatiques.

Le jardinier :

Il s’occupe de la serre et des plantations et agit en conséquence des mesures climatiques relevées par les capteurs.

# Intégration et versionning

En travaillant tout les trois sur plusieurs logiciels similaires, nous avons décidé d’utiliser un dépôt privé grâce à l’utilitaire GitHub. Cet utilitaire nous permettra de nous simplifier grandement la tâche. Il suffit d’exporter le dossier du projet initial vers notre dépôt privé, puis de le cloner sur chacun de nos ordinateurs, que ce soit ceux de l’école ou nos ordinateurs personnels.

Durant la réalisation de notre projet, après chaque avancement dans une tâche, chacun des membres pouvait enregistrer ses modifications et les exporter vers le dépôt en ligne. Ainsi, nous avions tous en permanences les fichiers à jour du travail des uns et des autres. Cela permettait une intégration continuelle et un gain de temps considérable.

Github propose un utilitaire Windows nommé GitHub Desktop qui permet d’effectuer tout ces opérations avec facilité. Chaque modification, ajout ou suppression de fichier sont notifié dans un historique de manière que les modifications sont consultables, et réversibles. Les lignes de codes modifiés apparaissant.

Cependant, bien que GitHub nous soit extrêmement utile, nous avons rencontrés quelques problèmes de conflits. En effet, si plusieurs étudiants modifient le même fichier en même temps, la fusion des ajouts entrainent des conflits. Mais nous avons pu surmonter ces problèmes très facilement.



# Diagramme

## **Diagramme d’exigences**



Le diagramme d’exigences met en évidence plusieurs points qui font écho au diagramme de cas d’utilisation.

Afin de pouvoir être informé des évènements de la serre, un utilisateur doit être en mesure de visualiser des informations sur l’état du système sur une courte période via une application Android, les informations en temps réel, ainsi qu’une visualisation d’état du système sur une période variable sous forme de graphique sur un interface Web. Toutes les informations d’évènements importants seront journalisées et horodaté afin de mieux être averti en cas de problème sur le système.

Pour être correctement préparé pour ces actions, le système doit être en mesure de mémoriser toutes les mesures du système et cela de façon fiable et les mesures seront prises par intervalle facilement paramétrable par l’exploitant.

## **Diagrammes des cas d’utilisation**

Cas d’utilisation : Visualiser l’évolution d’une mesure



A chaque connexion de l’utilisateur sur le site, les relevés des capteurs sont mis à jour en temps réel en lien avec la base de données via des requêtes SQL en PHP.

Le choix de période de temps sera représenté par des graphiques montrant l’évolution de ces mesures dans le temps.

L’ajout de capteur est une fonction automatique qui permettra au site de s’adapter automatiquement à l’ajout d’un nouveau capteur dans le système. C’est-à-dire sans que l’utilisateur ai à changer le code source d’en récupérer la valeur et l’afficher.

## **Diagrammes de séquences**

Description : Ce diagramme projette les interactions entre la carte de gestion et la base de données en fonction du temps.

Une image contenant carte, texte

Description générée automatiquement

Description :

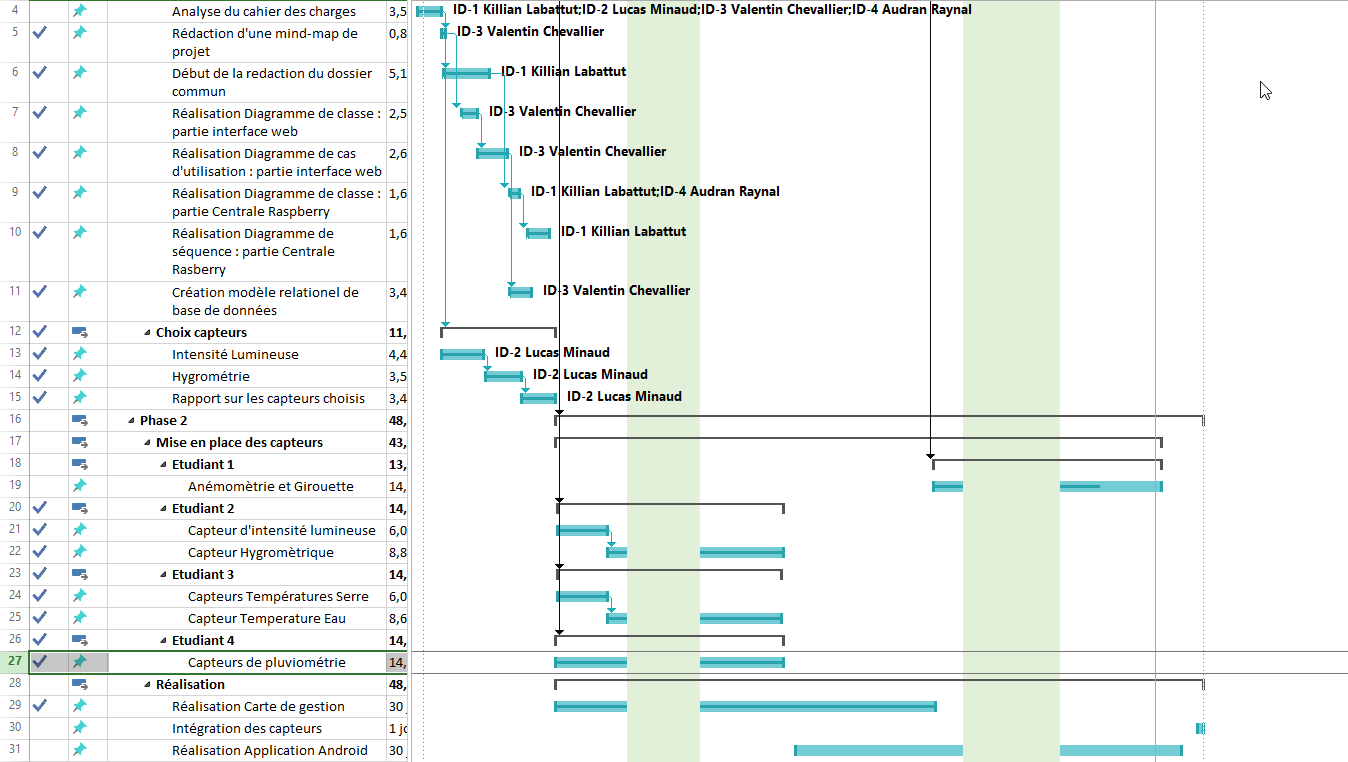
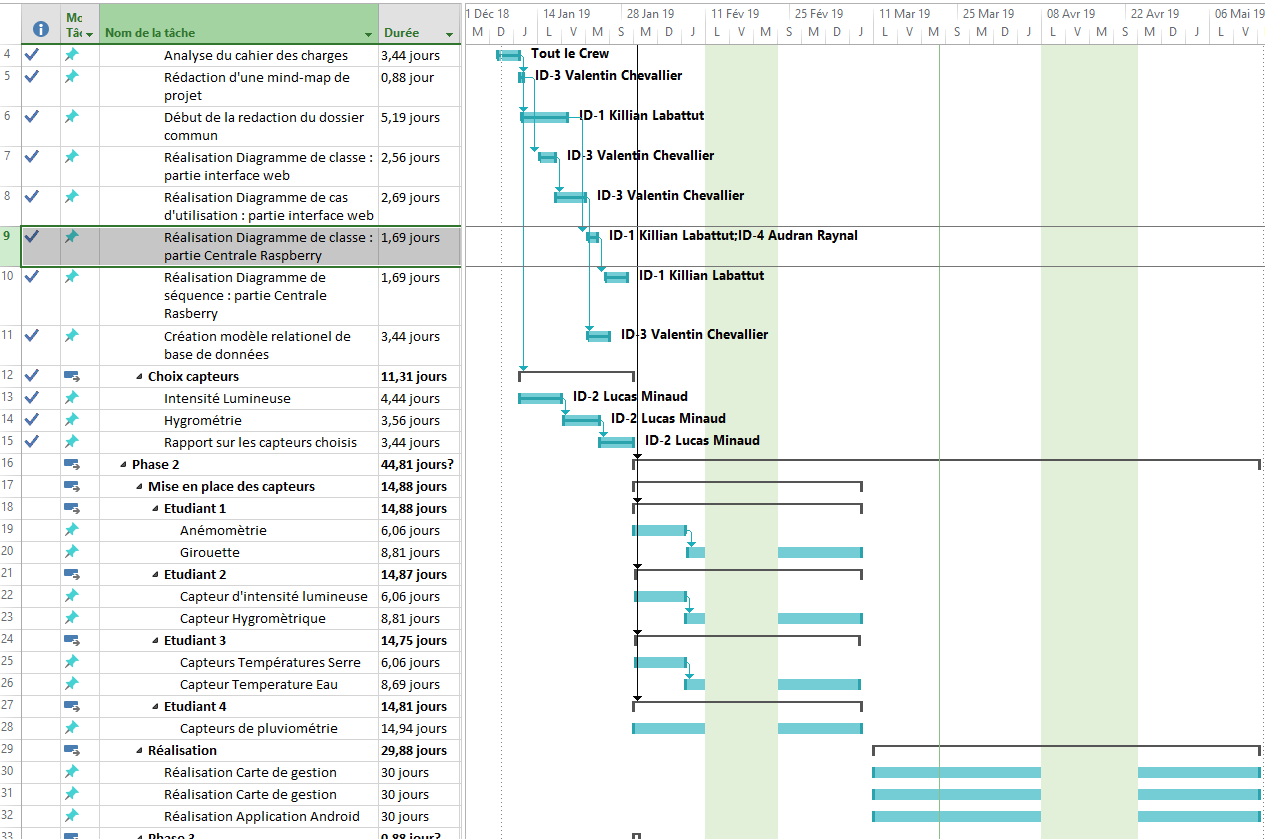
Une image contenant capture d’écran

Description générée automatiquement

## **Diagrammes de classes**



## **Diagramme de Gantt**



Voici deux diagrammes de Gantt distincts, l’un date du mois de Mars et l’autre du mois de Mai. Ce diagramme nous sert à visualiser l’avancement général du projet dans sa continuité. En effet on peut constater l’avancement des tâches entre les deux dates. Les tâches sont nommées et datées dans un ordre chronologique.

## **Diagramme de Base de données**



VC

Etant donné que les deux groupe (SFL5 et 6) partagent la même base de données

Le projet SFL5 utilise uniquement la table réservée à ce dernier :

Capteur :

Contient le nom du capteur ainsi que la période d’acquisition, c’est-à-dire une valeur de temps ou l’application saura quand effectuer chaque mesure (ex : il ne sert à rien de relevé la température chaque seconde). Ainsi que son état (en fonctionnement ou non).

Relevé :

Contient la grandeur physique mesurée et un format DATETIME pour indiquer la date et l’heure a laquelle a été mesuré cette valeur.

## **Diagramme de Déploiement**



**Mesure de la température :**

Les deux capteurs étant des sondes RTD c’est-à-dire des capteurs qui mesure la température de manière analogique avec la résistance, L’utilisation d’une boucle 4-20 Ma est obligatoire pour numériser la valeur captée.

Grace au Shield Libellium pour Arduino on peut récupérer la valeur de tension lié à la résistance donc ici pour le capteur de température de l’eau on obtient 4 Ma pour 0 ° C et 20 Ma pour 100 °C.

Une alimentation extérieure est donc nécessaire et après des recherches sur les capteurs ont connais l’alimentation qui est de 24 v.