

Cahier des Charges

Cloud of Things



Table de matières

Table de matières	1
Concept	2
Contexte du projet	2
Problématique	2
Ambitions	3
Clients	3
Architecture	4
Objectifs	4
Avantages de notre solution	5
Limites	5
Choix technologiques	6
Côté serveur	6
MongoDB	6
Mosquitto -MQTT Broker	6
Flogo	6
Middelware	6
NODE.JS	6
Express js	7
Côté Client	7
Ionic	7
Angular	8
Capacitor	8
Carte utilisé	8
Capteurs Nécessaires	8
Modèles commerciales	9
4P Marketing Matrix	9
Livrables	10
Contraintes	10
Avantages et inconvénients de l'extreme programming :	15

Concept

Contexte du projet

De nos jours, la population mondiale ne cesse pas d'augmenter d'une façon assez rapide avec des expectations de passer de 7,7 milliards d'individus aujourd'hui à 9,7 milliards en 2050. Cette augmentation a certainement un effet péjoratif sur les ressources alimentaires surtout que les terrains sont en train de diminuer progressivement et que leur pourcentage ne dépasse pas 11 %.

La nécessité d'augmenter la production alimentaire malgré ces contraintes pose une problématique remarquable. Cependant, l'émergence des technologies avancées touchant tous les domaines de vie(santé, industrie, énergie....) présente une solution efficace pour le secteur d'agriculture. L'internet des objets s'est introduit dans tous les domaines et tous les secteurs en particulier avec l'émergence des technologies LPWAN (Low Power Wide Area Network) moins énergivores. L'IoT semble d'ailleurs d'une grande aide au secteur agricole. Dans un contexte régi par les défis inhérents au changement climatique et l'appauvrissement en ressources naturelles à l'échelle mondiale (pénuries d'eau, appauvrissement des sols, accroissement des émissions de gaz à effet de serre, poursuite de la déforestation et de la dégradation des terres agricoles), l'agriculture est l'un des secteurs qui requiert une mobilisation générale.

L'un des plus grands domaines d'application de l'Iot dans le secteur de l'agriculture est le Smart Greenhouse : il s'agit dans ce cas de surveiller les conditions climatiques en permanence dans une serre, de détecter des variations et de mettre en œuvre des actions correctives pour maintenir des conditions optimales pour la croissance des plantes.

Dans notre travail, nous proposons de concevoir une solution intelligente pour les serres dite "Smart Greenhouse" permettant d'assurer une productivité optimale.

Problématique

La serre , une approche d'agriculture urbaine, offre aux agriculteurs la possibilité de fournir des conditions de culture optimales en offrant un environnement contrôlé en fonction des

exigences de la culture. Cependant, de nombreux agriculteurs ne parviennent pas à obtenir de bons bénéfices et le rendement souhaité des cultures en serre, car ils ne peuvent pas surveiller et contrôler efficacement des facteurs importants tels que la lumière, l'air, la température, etc., qui déterminent la croissance et la productivité des plantes.

Ambitions

- Éviter le risque de se faire voler en déverrouillant la porte de la serre et en se basant sur la localisation de mobile et la technologie NFC .
- Contrôle efficace de température, de l'humidité et de la pression au sein de chaque serre en implémentant des solutions efficaces d'aération et de constitution d'ombre.

Clients

- Le marché potentiel : le marché agricole tunisien.
- Le profil de la clientèle cible : les agriculteurs

Architecture

Ce système d'irrigation de serre intelligent est l'un des systèmes basés sur des capteurs qui peuvent fonctionner confortablement et amicalement dans l'agriculture en chaque serre. Il contient les différents types de capteurs pour différents potentiels comme : capteur de pression atmosphérique, capteur d'humidité du sol et capteur de température du sol. Notre architecture prend le modèle suivant décrit ci-dessous :

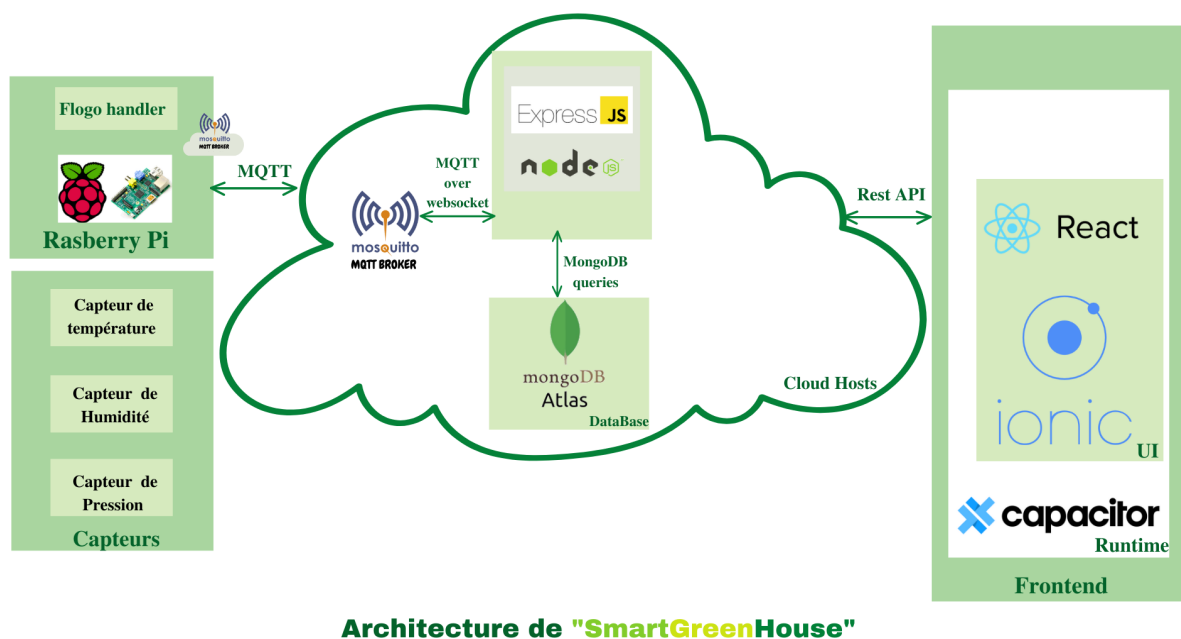


Figure 1 : Architecture de "SmartGreenHouse"

Objectifs

Notre solution SmartGreenhouse offre un contrôle avancé du microclimat et une optimisation énergétique.

Les producteurs peuvent surveiller et contrôler les paramètres mentionnés ci-dessous pour assurer un meilleur taux de croissance de la culture :

- Gestion de la pression au sein de la serre
- Acquisition de la température
- Gestion de l'humidité du sol
- Déverrouiller la porte de la serre en se basant sur la localisation de mobile et la technologie NFC

Donc les producteurs peuvent surveiller les paramètres suivants pour comprendre le cycle de croissance des plantes et prendre des mesures proactives en recevant des alertes si l'un des facteurs est affecté.

Avantages de notre solution

-Gérer l'ouverture et la fermeture des portes des serres

Grâce à notre solution, l'agriculteur peut verrouiller ou déverrouiller les portes des serres pour se protéger contre les vols des cultures agricoles.

-Créer le bon environnement pour un meilleur rendement

Avec Smart Greenhouse, les agriculteurs peuvent adapter un environnement pour leurs cultures qui offre une atmosphère intelligente face au climat et sensible à la nutrition pour augmenter la qualité des cultures.

-Surveiller, contrôler et détecter la croissance des plantes

Notre solution Smart Greenhouse permettra aux producteurs de surveiller leurs paramètres essentiels à une croissance saine de la culture, d'envoyer des alertes en cas de problème et de tout gérer à distance sur n'importe quel appareil.

-Analyse prédictive permettant de réduire les coûts

Limites

L'acheminement des données de différentes serres passe toujours par le cloud, ce qui dépend fortement du réseau Internet utilisé . Un mauvais débit d'internet peut résoudre en un mauvais suivi des paramètres de notre serre .

Nous avons besoin nécessairement d'un réseau performant et permettant le transfert de données avec un délai de retard négligeable pour assurer le bon fonctionnement de notre solution.

Choix technologiques

Côté serveur

MongoDB

MongoDB Atlas est la base de données globale en version Cloud pour les applications modernes. Elle est distribuée et sécurisée par défaut, et disponible en tant que service entièrement géré sur AWS, Azure et Google Cloud.

Mosquitto -MQTT Broker

Mosquitto est un serveur MQTT Open Source (Broker) qui facilite la communication entre objets connectés (M2M).

Flogo

Flogo gère les données collectées par les capteurs en fonction d'événements spécifiques

Middelware

NODE.JS

Node.JS, une plateforme qui s'est créée une place parmi les géants du web. Notre choix est justifié par plusieurs raisons qui sont :

- C'est du JavaScript, c'est simple et rapide.

En effet, avec Node.JS, JavaScript n'est plus seulement utilisé du côté client. Node.JS offre un environnement côté serveur qui prend la place de PHP, Java EE et d'autres langages serveur pour générer des pages web.

- Adapté aux RTA et aux SPA

RTA : Real Time Applications. Ce sont des applications qui doivent être mises à jour très fréquemment.

SPA : Single Page Applications. Ce sont des applications qui incluent une seule page html dont le contenu change en fonction des actions de l'utilisateur.

Node.JS nous permet donc de créer un code dans un modèle non-bloquant. C'est-à-dire que les actions ne sont pas exécutées dans l'ordre de lancement des requêtes.

- Une grande communauté de développeurs

Node.JS rassemble une grande communauté de développeurs adorant le partage. D'où le nombre énorme de workshops, conférences et meet-ups à propos de Node.JS.

- Une technologie stable et éprouvée

Les géants les plus influents tels que Netflix, Uber, Ebay... Node.JS suit une politique très stricte concernant sa mise à jour et sa maintenance. Dans toutes les versions de Node.JS, on n'effectue pas de changements majeurs qui endommagent votre application. Node.JS est donc, le bon choix à faire sur le long terme.

Express js

Express.js est un framework pour construire des applications web basées sur Node.js. C'est de fait le framework standard pour le développement de serveur en Node.js.

Côté Client

Ionic

IONIC a rationalisé le développement d'applications multi-plateformes, apportant une touche élégante à l'espace des applications hybrides. Notre est justifié par le fait que :

- Open source, absolument gratuit

Un avantage non négligeable, surtout si l'on souhaite customiser un peu son travail, l'adapter à son contexte professionnel,...Gratuit, mais pas bradé pour autant. En effet, l'inconvénient d'un outil Open Source à parfois été son absence d'évolution et d'adaptation aux défis technologiques toujours plus importants dans le temps.

- Grande communauté utile

En plus d'une documentation déjà très riche et simple à prendre en main, le Framework fédère une très large communauté de développeurs enthousiastes.

- Coder une fois, déployer partout

C'est sûrement l'un des plus gros avantages de Ionic : la possibilité de développer votre application une fois, et la déployer sur plusieurs terminaux mobiles.

Angular

Angular a été le premier framework qui a été pris en charge par le framework Ionic. Il permet de créer des applications mobiles multiplateformes pour Android et iOS en utilisant les composants Ionic et le code Angular. D'autres frameworks comme ReactJS et VueJS sont également pris en charge, mais Angular est toujours largement utilisé avec le framework Ionic pour le développement d'applications mobiles.

L'avantage de Angular par rapport React est la disponibilité des packages.

En effet, il existe un énorme référentiel de packages open-source disponibles pour les développeurs Angular comme sont NgBootstrap, Angular Google Maps, NgRx, NgTranslate, AngularFire, NgxTextEditor, Angular Material, Ng2 Pdf Viewer, NgxCharts,....

Ces packages nous permettent de coder d'une manière flexible les applications mobiles.

Capacitor

Capacitor est un environnement d'exécution natif open source pour la construction d'applications Web natives. Créez des applications Web multiplateforme iOS, Android et progressives avec JavaScript, HTML et CSS.

Carte utilisé

Raspberry Pi

Le Raspberry Pi 4 est un nano-ordinateur qui se présente sous la forme d'une carte de processeur. Il peut se brancher directement sur un écran ou un moniteur.

Le Raspberry Pi 4 possède toutes les connectiques d'un PC de bureau de taille standard. Toujours plus puissante, cette quatrième génération permet de naviguer sur Internet, regarder des vidéos en 4K, faire des traitements de texte, voire jouer à certains jeux vidéo

Capteurs Nécessaires

- Capteur de pression
- Capteur d'humidité du sol
- Capteur de température du sol

Modèles commerciales

4P Marketing Matrix

Avoir un produit qui corresponde en tout point à la réalité du marché est un exercice complexe. Les habitudes des consommateurs évoluent et il est parfois compliqué de suivre le rythme du changement au sein d'une entreprise.

Le marketing-mix ou les 4P est VOTRE RECETTE pour faire en sorte que vous atteignez les objectifs que vous vous êtes fixés:

Politique Produit : Notre Produit sera le centre de notre stratégie Marketing.

Marque : Solution "Smart Greenhouse" qui combine deux parties hardware et software qui se résume en une application hybride.

Services liés au produit :

- Service Après-Vente.
- Des mises à jour, des nouvelles fonctionnalités et des améliorations quotidiennes.
- Des garanties allant jusqu'à 3 ans.

Caractéristiques: Diversification et variété en termes de fonctionnalités, options, design..

Politique Prix : Le prix est au cœur du positionnement et l'élément clé de la communication.

La politique suivie est la politique d'écémage suite à l'absence d'une concurrence concrète à la Tunisie. Le prix de la solution dépend des fonctionnalités que le client souhaite implémenter dans sa serre.

La politique de réduction commerciale :

- Des remises et des promotions auront lieu s'il y a plus que 3 fonctionnalités à implémenter.
- Les modalités de paiement varient de virement bancaire ou paiement online.
- Réductions exceptionnelles accordées aux clients pour cause de problème de qualité ou de non conformité

Les modalités de paiement varient de virement bancaire ou paiement online.

Politique Distribution: La distribution et le choix de son canal conditionne la visibilité et l'accessibilité du produit.

Canaux de distribution: La livraison directement chez nos agriculteurs.

Politique Communication : Nous proposons toutes les techniques publi-promotionnelles destinées à soutenir un produit.

- Mesures publicitaires hors ligne: télévision, radio, journaux, magazines, affiches, etc.
- Mesures publicitaires en ligne: Display Advertising, Vidéos (sur YouTube, Facebook, etc.), publi- cité sur les réseaux sociaux, etc.
- Communication personnelle : Échanges avec le client.
- Communication internet : E-mailings, Newsletters, Marketing des médias sociaux.
- Les Ventes : Démarchage, Démonstration, Participation à des salons et foires...
- La Publicité : radio, presse, journaux, affichage, brochures, plaquettes,
- Les Relations Publiques : communiqués et dossiers de presse, opérations de sponsoring ou de mécénat, parrainage, lobbying, tenue d'événements.

Livrables

- **Cahier conceptuel** : Ce document présente de manière détaillée et structurée les spécifications, les services à rendre, les contraintes de cette
- Solution aussi que la conception architecturale et la conception détaillée.
- **Exécutables et Sources** : L'ensemble des instructions et des fichiers dans un répertoire en Github contenant le code de la solution IoT et de l'application mobile développée.
- **Documentation technique** : La totalité des bibliothèques et technologies utilisées dans le développement de cette solution ainsi que les références utilisées.
- **Vidéo de démonstration** : Une vidéo sous format mp4 qui contient une démonstration de la solution proposée.

Contraintes

Méthodologie de travail

On va travailler avec Extreme Programming (XP) qui est un cadre de développement logiciel agile qui vise à produire des logiciels de meilleure qualité, XP est le plus spécifique des cadres agiles concernant les pratiques d'ingénierie appropriées pour le développement de logiciels.

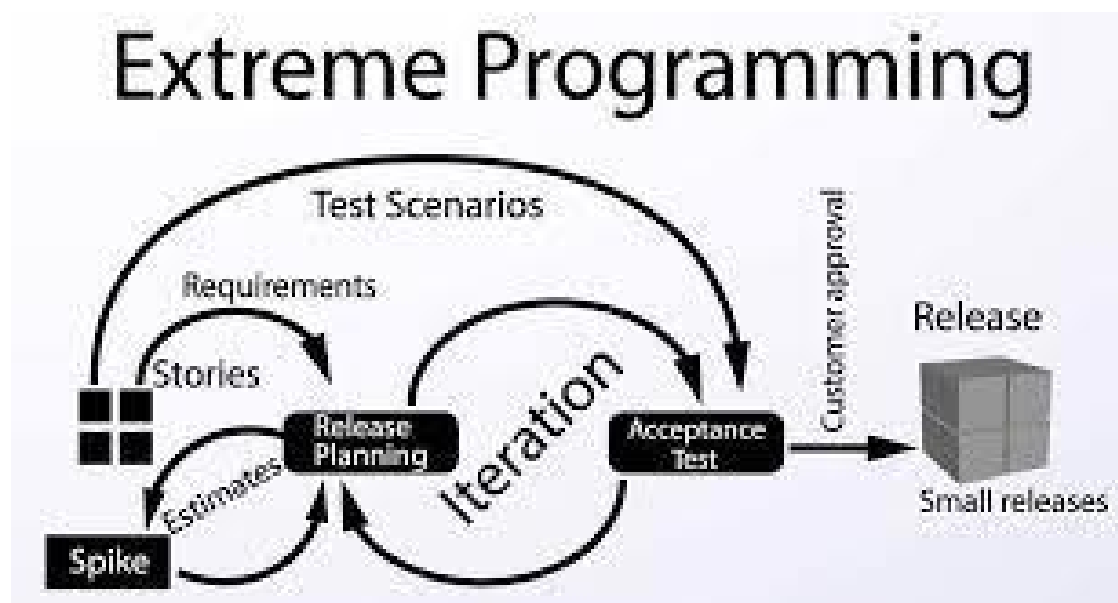
l'Extreme Programming est axé sur les exigences du client. Une évidence à priori ! Le développement de logiciels classiques ne peut pourtant répondre aux désirs des clients que dans une certaine mesure, les choses se compliquent notamment quand ces désirs changent régulièrement. XP tente par ailleurs de stimuler la créativité des développeurs et accueille les erreurs comme un facteur évident dans le processus de travail.

Principe :

Les principes de la méthode XP ne sont pas nouveaux puisqu'il s'agit de ceux des méthodes agiles. La différence et l'originalité résident dans le fait qu'elles sont poussées à l'extrême.

La méthode XP s'appuie sur :

- Une forte réactivité au changement des besoins du client
- Un travail d'équipe
- La qualité du travail fourni
- La qualité des tests effectués au plus tôt



Techniques

Les pratiques de XP sont des consignes et des méthodes de travail très concrètes. Alors que les valeurs et principes présentés sont également appliqués dans d'autres méthodes de travail agiles, les techniques concrètes de l'extreme programming sont des caractéristiques uniques. Elles ont aussi légèrement évolué au fil du temps et sont différentes d'une source à une autre. De manière générale, les techniques se divisent en quatre domaines différents.

Feedback détaillé:

Dans le cadre de l'extreme programming, les équipes de développeurs travaillent dans des cycles extrêmement courts. Cela permet de tester encore et encore le code écrit. Le test unitaire ou Test-Driven Development va même jusqu'à écrire un environnement de test avant la création du code source à proprement parler. Le code ne passant pas ce test ne peut pas continuer à être développé. Le feedback vient donc ici du même système.

Le Planning Game est une réunion organisée au début de chaque phase de développement. L'équipe et le client se réunissent pour parler du travail accompli, donner un feedback et discuter des fonctions à venir. Les tâches sont ensuite distribuées.

L'idée d'un client sur site ou On-Site Customer contribue également aux feedbacks réguliers. Dans le meilleur des cas, au moins un représentant du client doit intégrer l'équipe afin de répondre rapidement aux questions ou amener des idées et définir des priorités.

Enfin, le pair programming applique le principe des quatre yeux : deux personnes travaillent toujours simultanément au code. Pendant qu'un collègue écrit le code, l'autre vérifie le code source, fait des propositions d'amélioration et signale des erreurs. Cette méthode est certes très coûteuse car deux collaborateurs doivent être affectés simultanément à une seule tâche, mais le code créé est finalement meilleur et requiert moins de rectifications.

Processus continu

Les équipes XP révisent leur code en permanence. Ce remaniement du code ou refactoring doit veiller à améliorer le code source ainsi qu'à éliminer des répétitions et des composants inutiles. Un code optimisé de la sorte est plus compréhensible, même pour des lecteurs externes, et donc moins source d'erreur.

Dans le cadre de l'extreme programming et d'autres méthodes de travail agiles, l'intégration continue permet aux équipes d'intégrer en permanence le nouveau code dans l'ensemble du projet. Plusieurs fois par jour, un développeur intègre son travail dans le projet. Les différentes contributions sont ainsi contrôlées en continu et tous les intervenants travaillent avec le dernier état.

La livraison des programmes et des mises à jour opérationnelles se fait le plus tôt possible selon XP. Les petites livraisons ou Small Releases augmentent également la fréquence des feedbacks. Les erreurs peuvent ainsi être détectées plus rapidement et supprimées dès la mise à jour suivante. Le client a perpétuellement la possibilité de tester directement le dernier état du développement et de formuler des critiques et des propositions.

Compréhension commune

Avec une conception simple (Simple Design), le code est compréhensible pour tous les intervenants. Tout ce qui complique inutilement le code source doit donc être supprimé. Pour les développeurs qui travaillent suivant l'extrême programming, le but est d'éviter tous les doublons. L'objectif du programmeur concerné devrait par ailleurs ressortir clairement du texte source.

Pour que toute l'équipe puisse travailler main dans la main, des normes de codage ou Coding Standards sont en principe définies. Ces spécifications se rapportent à l'approche et au format. Les collègues doivent également pouvoir s'y retrouver dans le code des autres, et en même temps il faut toujours pouvoir retrouver qui a procédé à telle ou telle modification.

La possibilité de travailler ensemble sur le code renforce l'appropriation collective du code ou Collective Code Ownership : plutôt que d'indiquer qui est chargé de telle ou telle partie (et donc des erreurs qui y sont contenues), on considère le code comme un produit commun. Toute l'équipe en porte la responsabilité tant pour les erreurs que pour les réussites. Cette technique invite en outre à perfectionner le code d'un autre et à apporter ses idées.

Rôles :

Dans l'extrême programming, les rôles servent à répartir les tâches et les compétences entre tous les intervenants, tant les développeurs que les clients.

Client

L'extrême programming agit de manière très orientée client, ce qui peut aller jusqu'à intégrer le client comme un membre de l'équipe et à avoir au moins un représentant sur site (On-Site Customer). Le client pose ses exigences pour le produit, mais n'indique que partiellement la

façon d'atteindre les objectifs. Seule la hiérarchisation des secteurs partiels relève de son domaine de compétences. Il doit ici également réussir à faire comprendre ses propres souhaits.

Développeurs

L'équipe de développeurs n'est pas sous-divisée. Autrement dit : toute personne créant activement le produit prend la casquette de développeur. L'équipe regroupe donc non seulement les programmeurs, mais aussi d'autres personnes participant à la création, en fonction des exigences du projet. Outre le travail de développement effectif, la mission des développeurs est également de réagir aux désirs du client : évaluer la charge de travail, établir un calendrier, planifier la mise en œuvre.

Manager

Le rôle du manager consiste à faire le lien entre les développeurs et les clients. Les personnes de ce groupe amènent les deux autres à une même table et animent par exemple le planning game. Le manager veille ici à ce que les règles définies au préalable ainsi que les conventions générales d'une discussion constructive soient respectées. Le manager endosse donc si nécessaire un rôle de médiateur.

Son rôle est parfois également qualifié de tracker (traqueur). En effet, l'une des missions du manager est de noter les indices de performance clés (p. ex. le temps que chaque collaborateur consacre au projet).

Coach

Toute l'équipe (y compris le client) doit savoir gérer l'extrême programming et pouvoir mettre en pratique cette méthode de travail de manière cohérente. Un coach peut contribuer à ce que tous se fassent une même idée des procédures. Il n'a rien à voir avec le développement du produit à proprement parler et ne sert que d'aide externe, de manière très similaire à un Scrum Master. Dans les discussions préalables, les règles et les pratiques peuvent être passées en revue avec cette personne. Le coach accompagne l'équipe idéalement pendant toute la phase de développement, est disponible pour répondre aux questions et aide à clarifier des points obscurs.

Avantages et inconvénients de l'extreme programming :

Avantages	Inconvénients
Contact étroit avec les clients	Charge de travail supplémentaire
Aucun travail de programmation inutile	Le client doit participer à la démarche
Logiciel stable grâce à des tests continus	Charge horaire relativement élevée
Prévention des erreurs grâce au pair programming	Coûts relativement élevés
Aucune heure supplémentaire, propre rythme	Possible uniquement avec une gestion de la version
Les modifications peuvent être prises en charge à court terme	Requiert de l'autodiscipline pour la mise en œuvre
Code clair à tout moment	

Contrainte de temps

7 semaines ne seront pas suffisantes pour bien terminer le développement de cette solution avec toutes les spécifications nécessaires et obtenir une version stable de l'application.

Vices cachés

Une solution robuste doit contenir des tests unitaires pour assurer le fonctionnement parfait de l'application, on peut alors obtenir des vices cachés dans la solution développée si on n'arrive pas à couvrir tous les scénarios possibles avec les tests unitaires.

Manque d'expérience

Dû au manque d'expérience des membres d'équipes, on peut commettre des fautes dans l'estimation du temps nécessaire pour les tâches de développement.

Diagramme de GANTT

La méthode Gantt consiste à déterminer la meilleure manière de positionner les différentes tâches du projet à exécuter, sur une période déterminée, en fonction:

- Des durées de chacune des tâches
- Des contraintes d'antériorité existant entre les différentes tâches
- Des délais à respecter
- Des capacités de traitement

Le développement de notre solution sera sur 9 parties:

- Planification : 5 jours de 02/11/2021 à 08/11/2021
- Conception : 5 jours de 09/11/2021 à 15/11/2021
- Développement : 29 jours de 16/11/2021 à 24/12/2021

- Partie Hardware: 9 jours de 16/11/2021 à 26/11/2021
- Partie Software: 20 jours de 29/11/2021 à 24/12/2021
- Test: 5 jours de 27/12/2021 à 31/12/2021
- Déploiement : 4 jours de 03 /01/2022 à 06/01/2022
- Documentation : 3 jours de 07/01/2022 à 11/01/2022

	i	Task Name	Duration	Start	Finish	Predecessors
1		Planification	5 days	Tue 02/11/21	Mon 08/11/21	
2		Conception	5 days	Tue 09/11/21	Mon 15/11/21	1
3		<input type="checkbox"/> Développement	29 days	Tue 16/11/21	Fri 24/12/21	2
4		partie Hardware	9 days	Tue 16/11/21	Fri 26/11/21	
5		Partie Software	20 days	Mon 29/11/21	Fri 24/12/21	4
6		Test	5 days	Mon 27/12/21	Fri 31/12/21	3
7		Déploiement	4 days	Mon 03/01/22	Thu 06/01/22	6
8		Documentation	3 days	Fri 07/01/22	Tue 11/01/22	7

