**REPUBLIQUE TUNISIENNE**

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

**ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE TUNIS EL MANAR**

****

**FACULTEDES SCIENCES DE TUNIS**

**DEPARTEMENT DES SCIENCES DE L'INFORMATIQUE**

**RAPPORT**

De Projet de Fin d’Etude

**SUJET**

DEVELOPPEMENT D'UNE APPLICATION WEB D’ADMINISTRATION D’EQUIPEMENTS CISCO

**Présenté par :**

Ben Salem Amr Amine

**Encadré par :**

Karim Salem BIAT

Hamed Aouadi FST

**Organisme d’accueil :** BIAT

*Année Universitaire 2016/2017*

# *Remerciements*

*Au terme de ce projet, nous tenons à exprimer notre profonde gratitude et notre immense respect, à notre encadrant Mr. Hamed Aouadi, pour sa disponibilité, ses avis, son aide considérable ainsi que son soutien. Nous lui témoignons un grand respect pour ses qualités humaines et professionnelles.*

*Nos vifs remerciements et notre gratitude s’adressent à tous le personnel de la BIAT notamment Mr. Karim Salem qui nous a donné la chance de travailler dans son équipe et d’apprendre de ses qualités professionnelles.*

*Avec beaucoup d’égard, nous ne manquerons pas d’exprimer notre grande reconnaissance à tous les enseignants de la faculté des sciences de Tunis et plus particulièrement aux membres du jury.*

# Dédicace

*A MA FAMILLE, que ce travail soit l’expression de ma reconnaissance pour vos sacriﬁces consentis, votre soutien moral et matériel que vous n’avez cessé de prodiguer. Que dieu vous accorde santé, bonheur et longévité.*

*A MES AMIS, Maroua, Nihel, Lachichi, Farouk et Saïd, vous êtes toujours là à mes côtés pour m’aider et m’encourager.*

*Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers.*

Table des matières

[Introduction générale 1](#_Toc483793781)

[1. Contexte et présentation du projet 1](#_Toc483793786)

[2. Organisme d’accueil 1](#_Toc483793787)

[2.1. Organigramme de la BIAT 2](#_Toc483793788)

[2.2. Présentation du département du système d’information (DSI) 3](#_Toc483793789)

[2.3. Réseau informatique de la BIAT 3](#_Toc483793790)

[3. Problématique et solution proposée 4](#_Toc483793791)

[4. Organisation du rapport 4](#_Toc483793792)

[Conclusion 5](#_Toc483793793)

[Etude préliminaire 6](#_Toc483793794)

[1. Procédure actuelle 6](#_Toc483793795)

[2. Revue de quelques applications similaires 6](#_Toc483793796)

[2.1.Cisco prime infrastructure 6](#_Toc483793797)

[2.2.Nagios 7](#_Toc483793798)

[2.3. Etude comparative 8](#_Toc483793799)

[Conclusion 8](#_Toc483793800)

[Spécification et conception générale 9](#_Toc483793801)

[1. Approche méthodologique et ingénieries adoptés 9](#_Toc483793802)

[2. Spécification des besoins 10](#_Toc483793803)

[2.1.Spécification des besoins fonctionnels 10](#_Toc483793804)

[2.2.Spécification des besoins non fonctionnels 12](#_Toc483793805)

[2.3.Gestion du projet avec Scrum 16](#_Toc483793806)

[3. Conception 19](#_Toc483793807)

[3.1.Dictionnaire de données 19](#_Toc483793808)

[3.2.Description des associations et des cardinalities 20](#_Toc483793809)

[3.3.Diagramme de classes 21](#_Toc483793810)

[3.4.Architecture logicielle 22](#_Toc483793811)

[Conclusion 23](#_Toc483793812)

[Release: Network Manager 24](#_Toc483793813)

[1.Sprint1 24](#_Toc483793814)

[1.1.Backlog du sprint 1 24](#_Toc483793815)

[1.2.Vue technique 25](#_Toc483793816)

[2.Sprint2: 30](#_Toc483793817)

[2.1.Backlog du sprint 2 30](#_Toc483793818)

[2.2.Vue technique 30](#_Toc483793819)

[3.Sprint3 36](#_Toc483793820)

[3.1.Backlog du sprint 3 36](#_Toc483793821)

[3.2. Vue technique 36](#_Toc483793822)

[4.Sprint4 41](#_Toc483793823)

[4.1.Backlog du sprint 4 41](#_Toc483793824)

[4.2.Vue technique 41](#_Toc483793825)

[5.Maquette 45](#_Toc483793826)

[5.1.Gabarit de la page d’authentification 45](#_Toc483793827)

[5.2.Gabarits des pages de l’application 46](#_Toc483793828)

[6. Burndown chart des quatre Sprints 49](#_Toc483793829)

[Conclusion 49](#_Toc483793830)

[Test et validation 50](#_Toc483793831)

[1. Évaluation du produit 50](#_Toc483793832)

[1.1.Scénario de test 50](#_Toc483793833)

[1.2.Validation de la qualité à la production 51](#_Toc483793834)

[1.3. Validation à l'utilisation 52](#_Toc483793835)

[1.4. Analyse de l’efficacité 55](#_Toc483793836)

[Conclusion 55](#_Toc483793837)

[Netographie 57](#_Toc483793838)

# Liste des figures

[Figure 1: L'organigramme de la BIAT 2](#_Toc483796046)

[Figure 2: Réseau de la BIAT (N3) 4](#_Toc483796047)

[Figure 3 Le tableau de bord de Cisco Prime Infrastructure 7](#_Toc483796048)

[Figure 4 Le tableau de bord de Nagios 7](#_Toc483796049)

[Figure 5: Cycle de vie Scrum 9](#_Toc483796050)

[Figure 6: Diagramme de cas d'utilisation 11](#_Toc483796051)

[Figure 7: Exigences du modèle square 12](#_Toc483796052)

[Figure 8: Modèle de la qualité au fonctionnement 15](#_Toc483796053)

[Figure 9: Plan de release 18](#_Toc483796054)

[Figure 10: Diagramme de classes 22](#_Toc483796055)

[Figure 11: Diagramme séquence système «s’authentifier» 26](#_Toc483796056)

[Figure 12: Diagramme séquence objet «s’authentifier» 27](#_Toc483796057)

[Figure 13: Première interface de l'application 28](#_Toc483796058)

[Figure 14: Interface de Login 28](#_Toc483796059)

[Figure 15: Tableau de tâches du Sprint 1 29](#_Toc483796060)

[Figure 16: Diagramme séquence système «Gestion des templates (Ajout)» 32](#_Toc483796061)

[Figure 17: Diagramme séquence objet «Gestion des templates (Ajout)» 33](#_Toc483796062)

[Figure 18: Interface d'accueil de l'application 34](#_Toc483796063)

[Figure 19: Interface de Templates 34](#_Toc483796064)

[Figure 20: Tableau de tâches du Sprint 2 35](#_Toc483796065)

[Figure 21: Diagramme séquence système «Backup des configurations» 38](#_Toc483796066)

[Figure 22: Diagramme séquence objet « Backup des configurations» 39](#_Toc483796067)

[Figure 23:Interface du backups et restores 40](#_Toc483796068)

[Figure 24; Interface de gestion des réseaux 40](#_Toc483796069)

[Figure 25: Diagramme séquence système «Programmer l’application à faire des taches programmées» 43](#_Toc483796070)

[Figure 26: Diagramme séquence objet « Programmer l’application à faire des taches programmées » 44](#_Toc483796071)

[Figure 27: Interface de Backups 45](#_Toc483796072)

[Figure 28: Gabrit de la page « d’authentification » 46](#_Toc483796073)

[Figure 29: Gabarit global de l’application 47](#_Toc483796074)

[Figure 30: Gabarit de la configuration des équipements 47](#_Toc483796075)

[Figure 31: Gabarit du backup et du restore de l’application 48](#_Toc483796076)

[Figure 32: Gabarit de la rubrique historique des informs et syslog 48](#_Toc483796077)

[Figure 33: Burndown chart 49](#_Toc483796078)

[Figure 34 Le déroulement du test 51](#_Toc483796079)

[Figure 35 : Résultat du test 52](#_Toc483796080)

[Figure 36: Questionnaire de satisfaction 54](#_Toc483796081)

[Figure 37: Evaluation de la satisfaction 54](#_Toc483796082)

# Liste des tableaux

[Tableau 1: Etude comparative des deux applications 8](#_Toc483796089)

[Tableau 2: Description des acteurs 10](#_Toc483796090)

[Tableau 3: Exigence du modèle SQUARE 14](#_Toc483796091)

[Tableau 4: Caractéristiques liées au fonctionnement 15](#_Toc483796092)

[Tableau 5: Backlog des Sprints 18](#_Toc483796093)

[Tableau 6: Dictionnaire des données 20](#_Toc483796094)

[Tableau 7: Backlog du Sprint1 24](#_Toc483796095)

[Tableau 8: Description textuelle de «s’authentifier» 25](#_Toc483796096)

[Tableau 9: Backlog du Sprint2 30](#_Toc483796097)

[Tableau 10: Description textuelle de «Gestion des templates (Ajout)» 31](#_Toc483796098)

[Tableau 11: Backlog du Sprint3 36](#_Toc483796099)

[Tableau 12: Description textuelle de «Backup des configurations » 37](#_Toc483796100)

[Tableau 13: Backlog du Sprint 4 41](#_Toc483796101)

[Tableau 14: Description textuelle de «Programmer l’application à faire des tâches» 42](#_Toc483796102)

# Introduction générale

## L

## e monde de l’informatique a connu une progression vertigineuse ces dernières années, ainsi la mise en réseau d’opérateurs distants est devenue un phénomène mondialisé impliquant plus d’un quart de l’humanité.

## La mise en réseau d’ordinateurs ou de terminaux parfois très éloignés géographiquement a permis de passer de l’ère industrielle, celui de la gestion limitée dans le temps et l’espace, à l’ère informationnelle offrant un cadre transnational et mondial.

## Un tel réseau planétaire n’aurait pas pu voir le jour sans une technologie et une infrastructure réseau informatique puissante, capable d’assurer des échanges à haut débit.

## Ces équipements ont des rôles spécifiques et variés, des technologies avancées, appartenant à des contextes et des sites très éloignés, ce qui pose alors le souci éminent de « la gestion à distance » de ces équipements.

## 1. Contexte et présentation du projet

Etant inscrit en troisième année de la Licence Fondamentale en Informatique à la Faculté des Sciences de Tunis, nous avons réalisé un projetde fin d’étude qui consiste en une application web intitulée « Network Manager » qui permet la supervision et la configuration centralisée des équipements réseaux (CISCO). C’est dans ce contexte que laBIAT nous a proposé un stage de quatre mois durant lequel nous avons conçu et développé cette application. Dans ce qui suit, nous présentons l’organisme d’accueil, la BIAT.

## 2. Organisme d’accueil

La Banque internationale arabe de Tunisie ou BIAT est une banque tunisienne qui fait partie des trente principaux établissements d'Afrique du Nord2. Première banque tunisienne en termes de parts de marché et cotée à la Bourse de Tunis (indice Tunindex 20), elle a pour actionnaire de référence le groupe Mabrouk.(N1)

La BIAT est créée en 1976 par la fusion des succursales locales de la Société marseillaise de crédit et de HSBC Bank Middle East (en). Cette création s'inscrit dans l'achèvement de la « tunisification » du système bancaire engagée au lendemain de l'indépendance.

***Fiche de présentation***

|  |  |
| --- | --- |
| **Date de constitution** | 01-févr-1976 |
| **Forme juridique** | Société Anonyme |
| **Registre de commerce No** | B 187 88 1996 |
| **Capital social** | 170.000.000 TND |
| **Effectif** | 2464 |
| **Nombre de clients** | 400 000 clients |
| **Nombre de comptes** | 540 000 comptes |
| **Nombre de points de vente** | 148 Agences locales et 1 à Tripoli en Libye 185 Distributeurs (DAB) |
| **Président de conseil d’administration** | Mr. Ismaïl Mabrouk |
| **Directeur général** | Mr. Mohamed AGREBI |
| **Réseau de correspondants internationaux** | 1400 banques |
| **Agence** | 200 Agences |

### Untitled-12.1. Organigramme de la BIAT

Figure 1: L'organigramme de la BIAT

### 2.2. Présentation du département du système d’information (DSI)

Le département du système d’information assure la sécurité et l’intégrité de l’information des clients de la BIAT ainsi que la sécurité et la rapidité de toutes les informations échangées lors des opérations. Ce département a pour tâche outre l’exploitation de l’outil informatique et la maintenance, aussi (N2) :

* Développer des applications informatiques.
* Gérer les ordinateurs du site central.
* Produire l’information sur les différents supports destinés aux clients ou aux services de la banque.
* Assurer en permanence la sécurité physique du site central.
* Assurer le fonctionnement du serveur vocal.
* Gérer les abonnés aux produits télématiques BIATOFIL.
* Concevoir les chéquiers et gérer leur diffusion aux agences via le bureau d’ordre central.
* Apporter une assurance technique à l’utilisation des produits télématiques et du serveur vocal aux clients.
* Préparer et mettre à la disposition des services des données et des statistiques sur l’utilisation de ses produits.

Le DSI de la BIAT est organisé selon quatre directions, à savoir :

* Direction maintenance, réseau et sécurité informatique.
* Direction exploitation informatique.
* Direction des études informatiques.
* Direction administration des systèmes et nouvelles technologies informatiques.

### 2.3. Réseau informatique de la BIAT

* Réseau de la BIAT

Le réseau de la BIAT est formé selon une organisation présentée par la figure 2

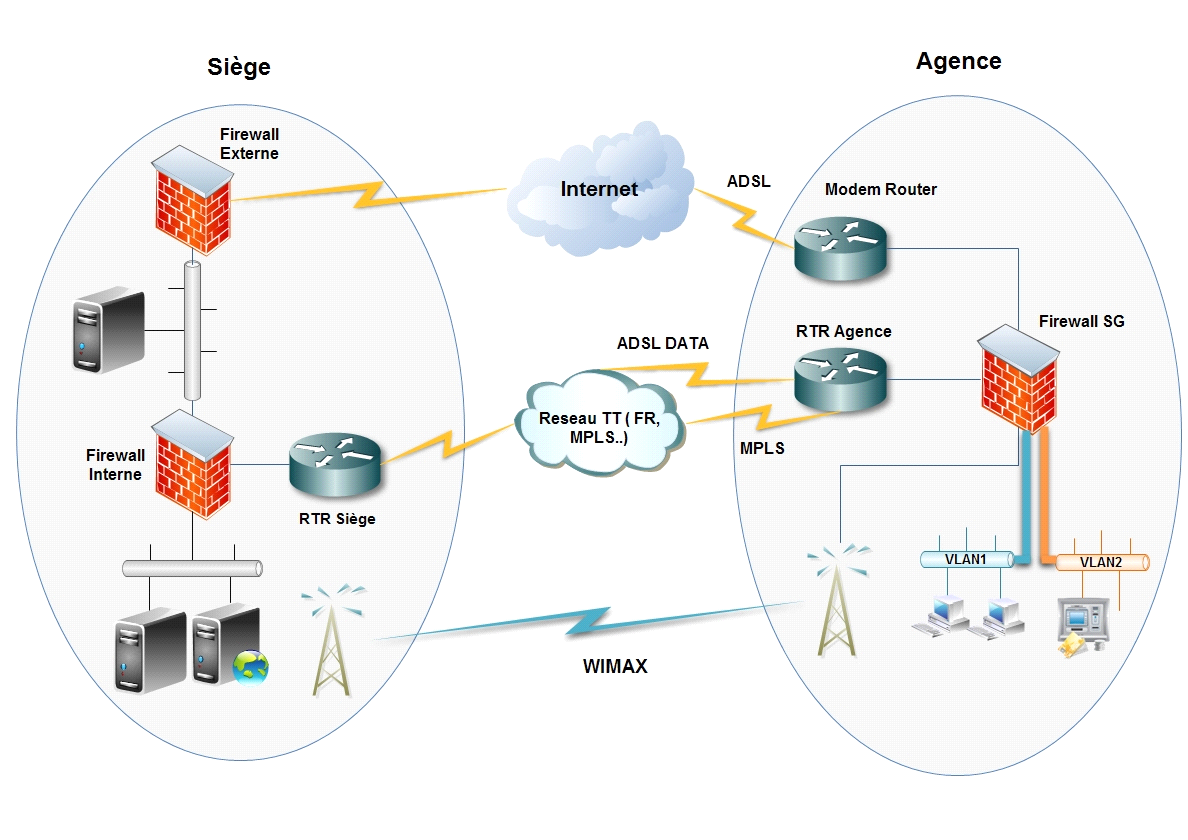


Figure 2: Réseau de la BIAT (N3)

## 3. Problématique et solution proposée

L'évolution des réseaux informatique pose un problème majeur : la manière dont ceux-ci sont pratiquement administrés. En effet, avec la multiplication des machines, la complexité des architectures qui s'en suivent, et leur possible éclatement géographique, il devient évidement difficile de gérer un réseau.

Le réseau de la BIAT souffre des mêmes problèmes. Dans le but de mieux gérer son réseau, nous proposons une application qui offre le moyen de :

* Superviser les réseaux;
* Configurer les équipements,
* Faire la sauvegarde et restauration des équipements.

## 4. Organisation du rapport

Le rapport est organisé comme suit.

* Le premier chapitre intitulé « étude préliminaire » met le projet dans son contexte général, présente l’étude de l’existant et décrit brièvement la solution que nous proposons.
* Le deuxième chapitre intitulé « spéciﬁcation et conception » présente les exigences du projet. Nous commençons par les besoins fonctionnels et non fonctionnels suivis par l’architecture générale.
* Le troisième chapitre « Release : Network Manager » met l’accent sur la phase de développement en respectant les règles de la méthodologie Scrum(N1) et en représentant ﬁdèlement les spéciﬁcations de l’application.
* Dans le dernier chapitre « Test et validation », nous décrivons la phase de test de l’application et du processus.

## Conclusion

Ce chapitre nous a permis d'introduire notre projet, de présenter l’organisme d’accueil et de préciser le travail demandé ainsi que l’organisation du rapport. Par ailleurs, nous allons nous intéresser, dans le chapitre suivant, à l’étude préalable du projet, qui est l’étude de l’existant, et ce, dans le but de mieux maîtriser le domaine de l’application à travers l’étude du processus actuel et de quelques applications similaires.

# Etude préliminaire

P

our la réalisation d’une application, une étude préliminaire est une phase primordiale qui permet l’analyse, l’évaluation et la critique des applications similaires pour arriver enfin à proposer une solution adéquate. Nous commençons, tout d’abord, de présenter le système actuel et son fonctionnement, ainsi que quelques applications similaires et leurs évaluations.

## Procédure actuelle

L’analyse de l’application actuellement utilisée à la BIAT (HP Open View) présente les caractéristiques suivantes :

***Avantages :***

- Elle permet une excellente supervision des équipements réseaux.

- Découverte automatique des équipements réseaux.

- Prend en charge plusieurs marques et modèles d’équipements.

***Inconvénients :***

- Ne permet pas la configuration des équipements,

- Ne permet pas de faire des sauvegardes et des restaurations des configurations des équipements (backups et restores)

En effet actuellement outre la lourdeur au niveau de la configuration des équipements, c’est-à-dire que l’agent chargé de le faire doit accéder d’une manière individuelle à chaque équipement. Le système actuel ne permet pas de faire des backups des configurations des équipements ainsi priver l’utilisateur de revenir en l’état en cas de panne ou de dysfonctionnement.

## 2. Revue de quelques applications similaires

Au cours de cette étape, nous nous intéressons à l’étude de deux applications similaires, Cisco prime infrastructure, Nagios.

### 2.1.Cisco prime infrastructure

Cisco Prime Infrastructure permet de gérer tout le cycle de vie des réseaux câblés et sans fil convergents. Ce produit regroupe les solutions qui permettent de simplifier la gestion des commandes et des licences.

La ﬁgure suivante illustre le tableau de bord de “Cisco prime infrastructure”.

Figure 3 Le tableau de bord de Cisco Prime Infrastructure

### 2.2.Nagios

Nagios (anciennement appelé Netsaint) est une application permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés, alertant lorsque les systèmes ont des dysfonctionnements et quand ils repassent en fonctionnement normal. C'est un logiciel libre sous licence GPL.(N5)

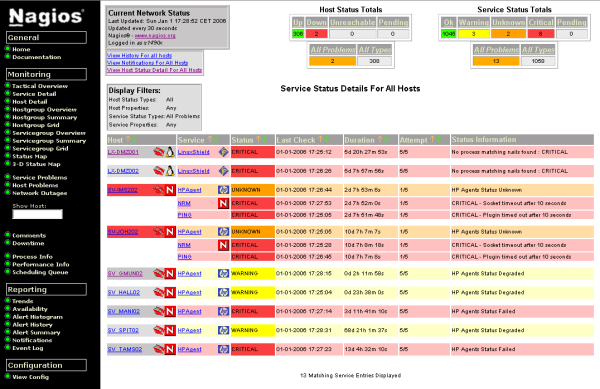
La ﬁgure ci-dessous illustre le tableau de bord de “Nagios”.

Figure 4 Le tableau de bord de Nagios

### 2.3. Etude comparative

Le tableau présente les points forts et les points faibles des applications étudiées dans la section précédente

|  |  |  |
| --- | --- | --- |
| **Caractéristiques** | **Nagios** | **Cisco Prime Infrastructure** |
| Payant | Gratuit/Payant | Oui |
| Supervision | Oui | Oui |
| Configuration | Non | Oui |
| Backup et restore | Non | Oui |
| Notifications | Avec plugins | Oui |
| Découverte automatique | Avec plugins | Oui |

Tableau 1: Etude comparative des deux applications

## 3. Environnement logiciel

Dans le tableau ci-dessous, nous allons présenter les logiciels utilisées pour le développement de la plateforme.

|  |  |
| --- | --- |
| **Logiciel** | **Description** |
| **C:\Users\Administrateur\Desktop\Nihel for the night\Sublime_Text_Logo.pngSublime Text 3** | Editeur de texte utilisé pour le développement de l’application |
| **Sybase Power Designer 16** | Environnement de modélisation pour les constructions des diagrammes UML |
| Easyphp_logo  **EasyPHP** | Serveur Apache |
| **Scrumwise** | Outil d’aide à la gestion de projet Scrum |
| **Photoshop CC 2014** | Logiciel de traitement et de dessin d’images |
| **Microsoft Word 2013** | Editeur de traitement de texte pour la rédaction du rapport |
| **Microsoft Powerpoint 2013** | Logiciel de présentation pour l’exposé |
| **BalsamiqMockups 3** | Outil de prototypage des interfaces |
| vmwareplayer_medium  **Vmware Workstation** | Permet la création d'une ou plusieurs machines virtuelles au sein d'un même système d'exploitation |
| Image associée  Putty | Permet de se connecter à distance à des équipements en utilisant les protocoles Telnet, SSH ou Rlogin. |
| Image associée  GNS 3 | Permet l'émulation ou la simulation de réseaux informatiques. |

Tableau 2: Environnement logiciel

### Technologies utilisées

Les différentes technologies adoptées et utilisées pour la réalisation de notre projet sont présentées dans le tableau ci-dessous:

|  |  |
| --- | --- |
| **Technologie** | **Description** |
| **Laravel 5.4** | Framework web open-source écrit en  PHP respectant le principe modèle-vue-contrôleur. |
| Image associée  **PHP 5** | Langage de programmation, utilisé pour produire des pages Web dynamiques. |
| **HTML 5** | Format de données conçu pour représenter les pages web. |
| **CSS 3** | Langage qui permet de gérer la présentation d'une page web. |
| **JavaScript** | Langage de programmation de scripts employé dans les pages web |
| C:\Users\Administrateur\Desktop\Nihel for the night\ajax-logo.pngAjax | Permet de construire des applications web et des sites web dynamiques interactifs. |
| Résultat de recherche d'images pour "jquery" JQuery | Bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. |
| Résultat de recherche d'images pour "bash"Bash | Interpréteur en ligne de commande de type script. |

Tableau 3: Description de la technologie utilisée

## Conclusion

Dans ce chapitre, nous avons fait une étude de l’existant afin de bien positionner la problématique. Cela nous a permis d’avoir des idées sur les apports potentiels de notre application. En vue de suivre une démarche logique dans ce rapport, une spécification des besoins fera l'objet du prochain chapitre.

# Spécification et conception générale

D

ans ce chapitre, nous traitons respectivement les phases de spécification et de conception.

## 1. Approche méthodologique et ingénieries adoptés

Après la collecte des données et la fixation des objectifs de notre projet, nous avons choisi Scrum (N6) comme méthodologie de gestion du projet.

Notre démarche de conduite du projet est basée sur les atouts de Scrum, il garantit une flexibilité dans la gestion du projet. Cette méthodologie nous a permis de répondre à toutes les exigences, en s’adaptant aux changements des besoins grâce à l’implication de celui-ci. Scrum ne couvre que les aspects de gestion de projet, et pour compléter le vide laissé en matière de pratique de développement, nous avons choisi de coupler Scrum avec XP(N7) (eXtreme Programming) qui est une méthode agile couvrant les bonnes pratiques d’ingénierie logicielle et le développement dirigé par le test.

Pour la spécification des besoins et la conception du projet, nous avons opté pour le langage de modélisation UML(N8) (Unifited Modeling Language). Notre choix s’est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu’il propose. UML présente aussi le meilleur outil pour schématiser des systèmes complexes sous format graphique et textuel simplifié.

La figure suivante présente le cycle de vie Scrum.

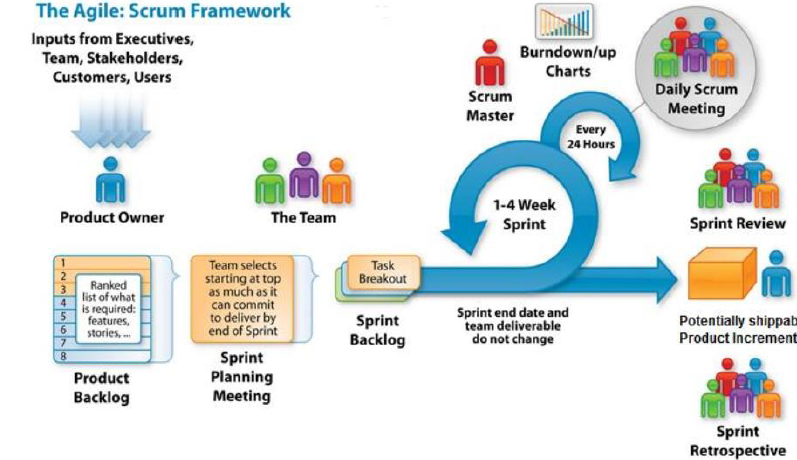


Figure 5: Cycle de vie Scrum

## 2. Spécification des besoins

### 2.1.Spécification des besoins fonctionnels

La spécification des besoins fonctionnels détermine ce que doit fournir le système en termes de fonctions utiles. C’est la phase de traduction des besoins utilisateurs en documentations conceptuelles et techniques.

Les besoins fonctionnels répondent aux points précis du cahier des charges, et sont donc requis par notre utilisateur final et lui sont indispensables. En d’autres termes, ce sont les besoins obligatoires ou encore les fonctionnalités principales de l’application.

#### 2.1.1.Description des acteurs

Dans ce paragraphe, nous identifions les acteurs du système. Pour ce faire, il faut discerner quels sont les différents rôles joués par les futurs utilisateurs. Il faut vérifier que les acteurs communiquent bien directement avec le système par émission et réception des messages.

* Supervisor
* Admin
* Super Admin

#### 2.1.2. Besoins par acteurs:

Dans cette partie nous allons définir le rôle de chaque acteur dans le système ainsi que les fonctionnalités que doit intégrer l’application à développer par acteur comme le montre le tableau suivant :

|  |  |
| --- | --- |
| Supervisor | -Supervision des réseaux. |
| Admin | -Supervision des réseaux.  -Configuration des équipements.  -Sauvegarde et restauration des configurations des équipements. |
| Super Admin | -Supervision des réseaux.  -Configuration des équipements.  -Sauvegarde et restauration des configurations des équipements.  -Gestion des utilisateurs. |

Tableau 4: Description des acteurs

#### 2.1.3.Diagramme de cas d’utilisation

Pour illustrer les fonctionnalités offertes par notre système, nous avons opté pour le diagramme de cas d’utilisation. Ce diagramme donne une vue sur les fonctionnalités de notre système ainsi que les acteurs qui l’utilisent.



Figure 6: Diagramme de cas d'utilisation

### 2.2.Spécification des besoins non fonctionnels

La qualité de notre application est considérée comme le facteur le plus important dans notre projet afin d'avoir un produit approprié offrant les meilleures fonctionnalités. La qualité d’un système se mesure par le degré auquel il répond aux besoins exprimés et implicites de ses différentes parties prenantes, et fournit ainsi une valeur ajoutée. Les besoins de ces parties prenantes

* fonctionnalité,
* performance,
* sécurité,
* maintenabilité...

Sont les critères de classification des produits, qui catégorisent la qualité du produit sur les caractéristiques et les sous-caractéristiques.

#### 2.2.1.Qualité du produit

##### 2.2.1.1.Modèle square

ISO 25000 est un modèle de qualité ayant pour objectif de poser le cadre et la référence pour définir les exigences qualité d’un logiciel (ISO 9126) et la manière dont ces exigences seront évaluées (ISO 14598).

Cette norme offre un cadre pour l’intégration des évolutions des normes ISO 9126, la qualité logiciel, et ISO 14598, l’évaluation de la qualité logicielle. En suivant les exigences du modèle Square nous nous intéressons à l’exigence liée à la production ainsi que l’exigence à l’utilisation. (N10)

##### 2.2.1.2.Exigences de la qualité liées à la production

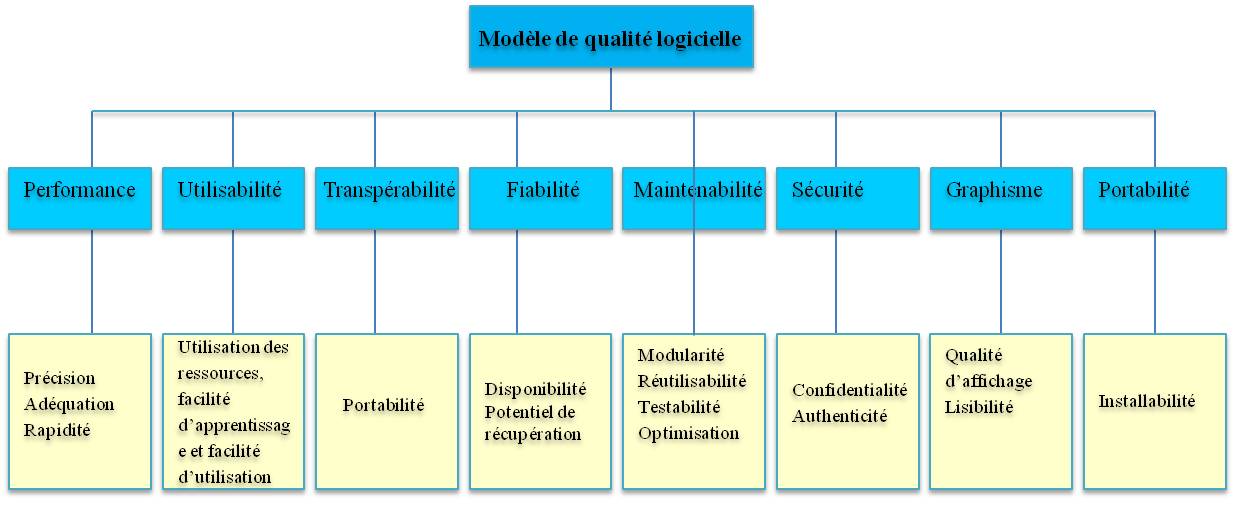
 Grâce à la norme ISO/IEC25000 nous pouvons spécifier les exigences de la qualité liées à l’utilisation. Le modèle de la qualité du produit comprend les huit caractéristiques de qualité liées à la production suivantes présentées par la figure ci-dessous:

Figure 7: Exigences du modèle square

Caractéristiques externes et internes : ce sont les caractéristiques évaluées au cours du processus de test en indiquant le poids de chacune :

* H : haut ;
* M : moyen ;
* F : faible ;

Le tableau suivant montre les caractéristiques retenus pour notre projet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caractéristiques externes** | | | | |
| ***Caractéristiques*** | ***Sous caractéristiques*** | ***Description*** | ***Poids*** | ***Métrique*** |
| Performance | Précision | Toutes les fonctionnalités fournissent le résultat attendu avec la précision attendue | H | Nombre d’interface non accessible |
| Adéquation | L’application fournit les fonctionnalités en adéquation avec les besoins spécifiés | H | Nombre de fonctionnalité existante et non utilisée par les utilisateurs finaux |
| Rapidité | Le temps de réponse d’une requête utilisateur doit être convenable | H | - Moyenne de temps en seconde pour se connecter  - Moyenne de temps en seconde pour effectuer une tâche |
| Utilisabilité | Utilisation des ressources | Les quantités et les types de ressources utilisées lorsque l'application exerce ses fonctions ne doivent pas alourdir l'application | H | Quantité des ressources utilisées |
| Facilité d’apprentissage | L’utilisateur doit apprendre comment utiliser l’application facilement | M | Nombre de fonctionnalités maitrisées après une période d’utilisation |
| Facilité d’utilisation | Les interfaces doivent plaire et satisfaire l'interaction de l'utilisateur. | M | Nombre de clics pour effectuer une tâche |
| Fiabilité | Disponibilité | L’application doit être disponible à tout moment | H | Temps nécessaire entre panne et reprise de fonctionnement |
| Potentiel de récupération | Nous aurons la possibilité de récupérer les données affectées et ré-établir l’état désiré si une interruption ou une panne se produit. | M | Possibilité de récupérer un état après une erreur |
| Transférabilité | Portabilité | L’application doit être compatible sur différents écran de différente taille (Smartphone, tablette) et doit avoir le même affichage | F | Possibilité de l’utilisation de l’application sur différents terminaux de la même manière |
| Sécurité | Confidentialité | Les informations confidentielles de l’utilisateur doit être sécurisées | H | Probabilité de trouver la clé de chiffrement |
| Authenticité | L’accès à l’application doit être précédé par une authentification | H | - Résistance au divulgation non autorisée d'informations  - Taux de succès en authentification avec des données érronés |
| Graphisme | Qualité d’affichage | Les images doivent être de bonne qualité | F | Taux de nombre d’image de bonne qualité |
| Lisibilité | La fonte doit être lisible | M | Pourcentage de police lisible |
| **Caractéristiques internes** | | | | |
| Maintenabilité | Modularité | Le code est composé de composants le moins couplés possibles | M | - Nombre de classes  - Nombre de méthodes |
| Réutilisabilité | Des fonctions peuvent être utilisées à construire un nouveau code | M | Quantité du code dupliqué |
| Testabilité | La facilité avec laquelle les critères d'essai peuvent être établis pour un système ou une composante. | H | Facilité de faire des tests multiples |
| Optimisation | Le code est optimal | H | Taux de l’optimisation du code |
| Portabilité | Installabilité | La facilité avec laquelle le produit peut être installé avec succès et/ou désinstallé dans un environnement spécifié. | H | Taux d'installations réussites |

Tableau 5: Exigence du modèle SQUARE

##### 2.2.1.3.Exigences de la qualité liée au fonctionnement

Cette partie spécifie les exigences de la qualité liée au fonctionnement, ce sont les caractéristiques avec lesquelles nous allons évaluer le produit après l’utilisation

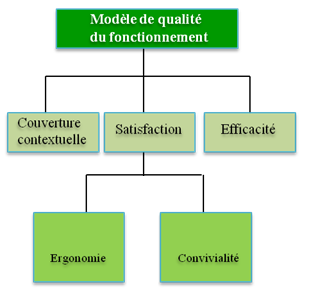


Figure 8: Modèle de la qualité au fonctionnement

Le tableau suivant montre les éléments de l’exigence de la qualité liée au fonctionnement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Caractéristiques*** | ***Sous caractéristiques*** | ***Description*** | ***Poids*** | ***Métrique*** |
| Satisfaction | Ergonomie | Facilité d'apprendre à utiliser l'application et la clarté des interfaces | H | Taux de difficulté durant l’utilisation de l’application |
| Convivialité | Capacité de l’application à être comprise, apprise, utilisée et attractive | H | Nombre de clics pour arriver à l’interface souhaitée |
| Efficacité | - | L'application doit répondre à toutes demandes de l'utilisateur | H | Le temps pris par un utilisateur pour trouver son besoin |
| Couverture contextuelle | - | L’application ne doit pas se planter ou perdre des données | M | Le nombre de plantage ou temps perdu au chargement |

Tableau 6: Caractéristiques liées au fonctionnement

### 

### 2.3.Gestion du projet avec Scrum

Travailler avec Scrum signifie respecter trois principales règles :

* + Utiliser les tableaux d’avancement des tâches, le découpage appelé «Backlog» du projet, ainsi que les «Backlog» de chaque sprint.
  + Respecter les différentes réunions de Scrum : la mêlée quotidienne, les planifications et revues de sprints ainsi que les réunions de rétrospectives.
  + Répartir les rôles :
    - le «Scrum Master» qui vérifie le respect des artefacts et réunions,
    - le «Product Owner» qui fait le Backlog et enfin, l’équipe de développeurs.
  + La planification sert à réaliser un projet bien organisé pour atteindre les objectifs souhaités, nous allons commencer tout d'abord par l’affectation des rôles SCRUM, ensuite la planification des réunions, le Backlog du produit et enfin le plan des livrables et des sprints.

#### 2.3.1.Affectation des rôles Scrum

Durant la phase projet, les rôles Scrum ont été affectés de la manière suivante :

* Le «Scrum Master» : c’est la personne qui vérifie le respect des réunions et l'avancement du travail selon le tableau de tâches : Dans notre projet c’est mon encadreur Monsieur Salem Thameur qui est le Scrum Master.
* Le «Product Owner» : C’est la personne découpant le projet en sprints et en livrables. Elle effectue ce que l'on appelle le «Backlog du produit». Elle est la responsable de définir et prioriser les users stories pour chaque sprint Ce sont les personnels du service informatique de la BIAT.
* L’équipe qui travaille chaque jour, commence sa journée par une mêlée quotidienne de mise au point, effectue le «Backlog du sprint» et réalise son avancement grâce au tableau des tâches. L’équipe est constituée par une seule personne Ben Salem Amr Amine.

#### 2.3.2.Backlog du produit

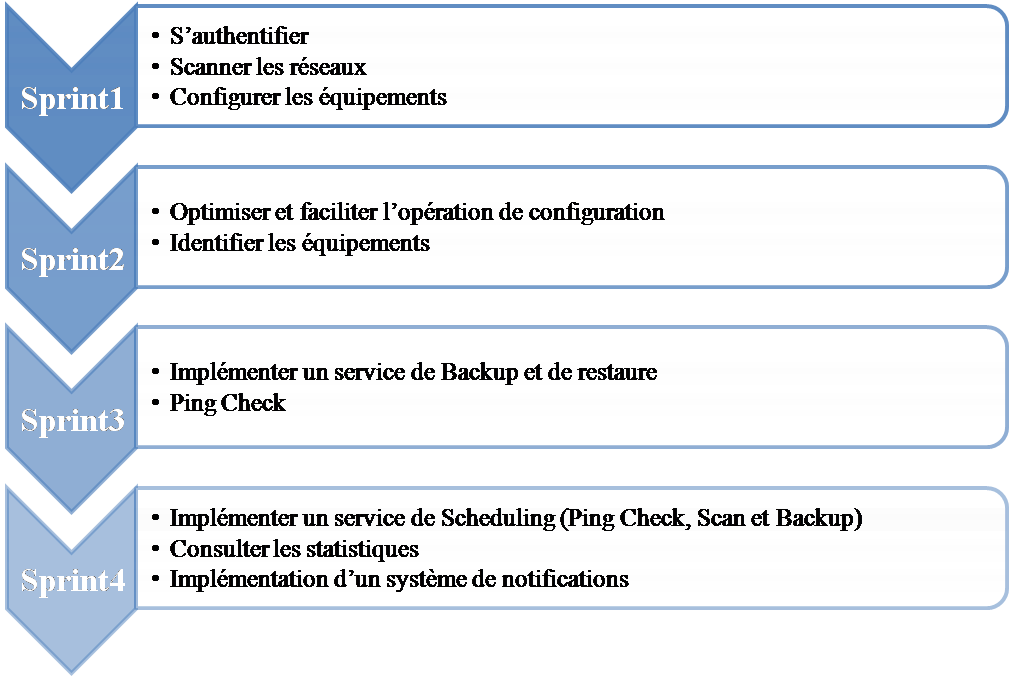
Le backlog du produit référence toutes les fonctionnalités que devra implémenter l’application. Il est présenté ici dans l’ordre de priorités définies par le client, et avec les estimations de l’équipe.Les éléments sont classés par priorité qui est généralement définie selon des calculs, mais nous avons choisi de simplifier ce calcul et de définir cette priorité par l’ordre de réalisation, c’est dire qu’une tâche prioritaire sera réalisée avant une autre qui est moins prioritaire. 1 jour de travail = 8 heures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***En tant que*** | ***Je souhaite*** | ***Afin de*** | ***Estimation*** | ***Sprint*** |
| Encadreur industriel | Avoir une application blanche | Tester les redirections et avoir un tableau de bord basique | 10 | Sprint 1 |
| Super Admin | M’authentifier | Accéder à mon espace de travail | 1 |
| Admin | M’authentifier | Accéder à mon espace de travail | 1 |
| Supervisor | M’authentifier | Accéder à mon espace de travail | 1 |
| Supervisor | Modifier les informations du compte | Changer l’émail et le nom | 1 |
| Admin | Scanner les réseaux | Avoir une liste des IP connectés et Filtrer les IP des équipements CISCO | 12 |
| Admin | Configurer les équipements scannés | Permettre la configuration de chaque équipement | 10 |
| Encadreur industriel | Tester le livrable | Valider le produit | 2 |
| Admin | Avoir un système de templates | Faciliter et optimiser l’opération de la configuration | 8 | Sprint 2 |
| Admin | Gérer les templates | Ajouter, modifier ou supprimer les templates | 4 |
| Supervisor | Avoir une liste des équipements et leurs informations | Consulter leurs informations et vérifier leurs états | 3 |
| Supervisor | Rechercher dans la liste des équipements | Trouver un/des équipements désirés | 1 |
| Supervisor | Avoir la liste des interfaces de chaque équipement | Consulter leurs informations et vérifier leurs états | 3 |
| Encadreur industriel | Tester le livrable | Valider le produit | 2 |
| Admin | Sauvegarder les configurations des équipements sur un serveur TFTP | Avoir un service de Backup | 4 | Sprint 3 |
| Supervisor | Consulter la configuration de chaque équipement | Avoir les configurations des équipements | 2 |
| Admin | Restaurer les dernières configurations sauvegardées | Mettre un équipement dans un ancien état | 4 |
| Supervisor | Gérer les réseaux | Ajouter ou supprimer des réseaux | 3 |
| Supervisor | Tester PING | Vérifier si les équipements sont connectés ou pas | 3 |
| Encadreur industriel | Tester le livrable | Valider le produit | 2 |
| Supervisor | Programmer l’application à faire des scans | S’informer périodiquement sur :   * L’état des équipements * Les nouveaux équipements * Créer un point de restauration pour les équipements | 15 | Sprint 4 |
| Supervisor | Être notifié des informs SNMP | Rester au courant des changements au niveau des équipements | 20 |
| Supervisor | Consulter des statistiques | Avoir une idée générale sur les réseaux | 3 |
| Supervisor | Consulter l’historique des informs et syslog | Avoir une idée les événements des informs et syslog | 1 |
| Supervisor | Rechercher un inform ou syslog dans l’historique | Trouver des informations facilement | 1 |

Tableau 7: Backlog des Sprints

#### 2.3.3.Planification des sprints et des releases

Dans le but de préparer le planning de travail et d’identifier le backlog des sprints, nous organisions une réunion de planification du sprint dans laquelle nous évaluons et repartionsles tâches qui doivent être réalisées durant ce sprint. Pour ce projet, nous choisissons de développer un seul release comme le montre la figure ci dessous.



**Release : Network Manager**

Figure 9: Plan de release

## 3. Conception

Après avoir fixé les besoins et les objectifs de l’application, nous procédons à la conception. Dans cette partie, nous allons tout d’abord décrire les architectures sur lesquelles notre système sera basé.

Ensuite, nous entamons la partie de la conception détaillée de l’application. Pour ce faire, nous présentons la vue statique du système à travers les diagrammes de classe et la vue dynamique via les diagrammes de séquence objet.

Enfin nous allons présenter la description détaillée des sprints.

### 3.1.Dictionnaire de données

Dans le tableau suivant, nous illustrons toutes les classes constituant la base de données, une description de tous les attributs qui figurent dans le diagramme de classes.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table*** | ***Code*** | ***Libellé*** | ***Type*** |
| Backups | Ip | Adresse ip de l’équipement | Varchar |
| Nname | Ip du Réseau de l’équipement | Varchar |
| Created\_at | Date de backup | Datetime |
| devices | Dip | Adresse ip de l’équipement | Varchar |
| Nname | Ip du Réseau de l’équipement | Varchar |
| Nlabel | Nom du réseau de l’équipement | Varchar |
| Sysref | Référence de l’équipement | Varchar |
| Syssoftware | Logiciel de l’équipeent | Varchar |
| Sysversion | Version de l’équipement | Varchar |
| Statuts | Status de l’équipement (connecté ou déconnecté) | Varchar |
| informs | Eventname | Nom de l’événement | Varchar |
| Eventid | Id de l’événement | Varchar |
| Community | Communauté ( pour snmp v2 ) | Varchar |
| Hostname | Equipement IP | Varchar |
| Agentip | Server IP | Varchar |
| Category | Catégorie de l’inform | Varchar |
| Severity | Sévérité de l’inform | Varchar |
| Uptime | Temps de disponibilité | Varchar |
| Traptime | Temps de réception de l’inform | Datetime |
| Formatline | Format de la ligne | Varchar |
| Seen | Status de la notification | Varchar |
| intdevice | Ip | Adresse ip de l’équipement | Varchar |
| Description | Description de l’interface | Varchar |
| Intip | Adresse ip de l’interface | Varchar |
| Type | Type de l’interface | Varchar |
| Mtu | Mtu de l’interface | Varchar |
| Speed | Vitesse de l’interface | Varchar |
| Adminstatus | Statut administratif | Varchar |
| Operstatuts | Statut opérationnel | Varchar |
| networks | nname | Nom du réseau | Varchar |
| Status | Etat du réseau | Varchar |
| Profile | Profil du réseau | Varchar |
| Pings | Ip | Adresse ip de l’ équipement | Varchar |
| Nname | Adresse ip du réseau | Varchar |
| Created\_at | Date de création du ping | Datetime |
| Status | Etat du ping | Varchar |
| profiles | Label | Nom du profil | Varchar |
| Pingnombre | Nombre de minutes/heures/jours | Int |
| Pingtime | minutes/heures/jours | Varchar |
| Scannombre | Nombre de minutes/heures/jours | Int |
| Scantime | minutes/heures/jours | Varchar |
| backupnombre | Nombre de minutes/heures/jours | Int |
| Backuptime | minutes/heures/jours | Varchar |
| restores | Ip | Adresse ip de l’équipement | Varchar |
| Nname | Réseau de l’équipement | Varchar |
| Created\_at | Date de création | datetime |
| Scans | Ip | Adresse ip de l’équipement | Varchar |
| Nname | Réseau de l’équipement | Varchar |
| Created\_at | Date de création | datetime |
| Status | Etat du scan | Varchar |
| templates | Title | Titre du template | Varchar |
| TFTP | Ip | Adresse ip du serveur TFTP | Varchar |
| Users | Name | Nom de l’utilisateur | Varchar |
| Email | Email de l’utilisateur | Varchar |
| Password | Mot de passe de l’utilisateur | Varchar |
| Role | Role de l’utilisateur | Varchar |
| Remember\_token | Utilisé pour le « Remember me » | Varchar |
| Created\_at | Date de création de l’utilisateur | Timestamp |
| Updated\_at | Date de modification de l’utilisateur | Timestamp |

Tableau 8: Dictionnaire des données

### 3.2.Description des associations et des cardinalities

Le diagramme de classes est basé sur les règles de gestion suivantes:

1. Un supervisor peut créer 0 ou plusieurs profils
2. Un Admin peut créer ou plusieurs profils
3. Un SuperAdmin peut créer 0 ou plusieurs profils
4. Un profil est créer par un seul utilisateur (Admin ou superAdmin ou supervisor)
5. Un superviseur peut ajouter 0 ou plusieurs réseaux
6. Un admin peut ajouter 0 ou plusieurs réseaux
7. Un SuperAdmin peut ajouter 0 ou plusieurs réseaux
8. Un réseau peut être ajouté par un seul utilisateur (Admin ou superAdmin ou supervisor)
9. Un admin peut restaurer 0 ou plusieurs équipements
10. Un SuperAdmin peut restaurer 0 ou plusieurs équipements
11. Un équipement peut être restauré par 1 ou plusieurs utilisateurs utilisateur (Admin ou superAdmin)
12. Un admin peut scanner 0 ou plusieurs équipements
13. Un SuperAdmin peut scanner 0 ou plusieurs équipements
14. Un équipement peut être scanné par 1 ou plusieurs utilisateurs utilisateur (Admin ou superAdmin)
15. Un admin peut faire un backup sur 0 ou plusieurs équipements
16. Un SuperAdmin peut faire un backup sur 0 ou plusieurs équipements
17. Le backup peut être effectué par 1 ou plusieurs utilisateurs utilisateur (Admin ou superAdmin)
18. Un équipement peut avoir 0 ou plusieurs backups
19. Un équipement peut avoir 0 ou plusieurs scan
20. Un backup peut être effectué sur un seul équipement
21. Un scan peut être effectué sur un seul équipement
22. Un équipement peut avoir 0 ou plusieurs pings
23. Un ping est associé à un seul équipement
24. Un équipement peut avoir 0 ou plusieurs informs
25. Un inform est associé à un seul équipement
26. Un équipement peut avoir 1 ou plusieurs interfaces
27. Une interface est associée à un seul équipement

### 3.3.Diagramme de classes

Le diagramme de classes est un diagramme statique d’UML, il décrit **les** structures des objets et des informations utilisés par notre application, à la fois en interne et en communication avec les utilisateurs. Il décrit également les informations sans référence à une implémentation particulière. Le diagramme présenté dans la figure suivante illustre les relations entre les classes de l’application « Network Manager ».



Figure 10: Diagramme de classes

### 3.4.Architecture logicielle

La conception de "Network Manager" est basée sur l'architecture MVC (Modèle-Vue-Contrôleur) (N10) qui est principalement basée sur la séparation des données (le modèle), d’interface homme machine (la vue) et de la logique de contrôle (le contrôleur).

Ce modèle de conception impose donc une séparation en 3 couches :

* Modèle : Définit l’interaction avec la base de données et le contrôleur ;
* Vue : Représente l’interface utilisateur. Elle se contente simplement d’afficher les données que lui fournit le contrôleur ;
* Contrôleur : Traite la requête de l'utilisateur pour lui fournir la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

L'utilisation de plateforme Laravel 5.4 (N11) nous a permet l’adoption du modèle MVC dans la conception de notre application. Laravel est basé sur la séparation entre les vue et les modèles et les contrôleurs.

Cette architecture présente de nombreux avantages parmi lesquels nous citons :

* La clarté de l’architecture qu’il impose ;
* La simplification du PHP.
* La simplification du développement lors d’une maintenance ou une amélioration sur le projet ;
* La modularité et la séparation de l’architecture qui nous offre la possibilité de travailler en parallèle la partie développement et la partie design ;

## Conclusion

Dans ce chapitre, nous avons mentionné la méthodologie adoptée, les besoins fonctionnels de l’utilisateur ainsi que les différents besoins non fonctionnels selon la norme Square pour préciser les qualités du produit. Ensuite nous passons à la phase de conception permettant d’aborder l’aspect technique des besoins en commençant par l’architecture de l’application

# Release: Network Manager

C

e chapitre présente notre application, nous avons eu recours a un découpage simple en sprints pour qu'on peut avoir un livrable exécutable, le release comporte 4 sprints.

## 1.Sprint1

Le sprint est un incrément qui constitue le cœur du Scrum.Nous organisons chaque semaine une réunion avec les membres de laBIAT où nous définissons nos buts à atteindre pour la semaine durant chaque sprint.

### 1.1.Backlog du sprint 1

Une réunion s’impose à chaque début du sprint pour planifier les tâches sélectionnées depuis le backlog du produit et leurs priorités à fin de concevoir un livrable à chaque fin du sprint.

* Date de début du sprint : 1 Février 2017
* Date de fin du sprint : 1 Mars 2017
* Temps estimé en heurs : 232h
* Echelle de mesure : Une journée de travail est égale à 8h de travail
* Objectif du sprint :
* Authentification des différents utilisateurs
* Scanner les réseaux
* Configurer les équipements

|  |  |  |
| --- | --- | --- |
| ***Element du backlog*** | ***Tâches*** | ***Estimation*** |
| Avoir une application blanche | Configuration du plateforme Laravel | 16h |
| Création de la base de données | 16h |
| Création d’un tableau de bord basique | 8h |
| Authentification | Développement de l’interface de l’authentification | 8h |
| Développement de l’authentification | 16h |
| Gérer les utilisateurs | Création des formulaires d’ajout et de modification des utilisateurs | 8h |
| Scanner le réseau | Développement de l’interface du Scan | 4h |
| Lister les IP connectées | 12h |
| Filtrer les équipements Cisco | 56h |
| Configurer les équipements | Développement de l’interface | 16h |
| Configuration des équipements | 56h |
| Tester le livrable | Tester l’authentification, le scan et la configuration des équipements | 16h |

Tableau 9: Backlog du Sprint1

### 1.2.Vue technique

Dans cette partie, nous élaborons la spécification des besoins fonctionnels, la conception dynamique et les interfaces du premier sprint.

#### 1.2.1.Spécification des besoins fonctionnels

Afin de spécifier les besoins fonctionnels du premier sprint, nous entamons en premier lieu la description textuelle puis les diagrammes des séquences système.

##### 1.2.1.1.Description textuel de «s’authentifier»

Avec l’aide de la description textuelle proposée par les concepteurs d’UML, les diagrammes auront une meilleure lisibilité

Le tableau suivant présente la description textuelle de «s’authentifier»

|  |  |
| --- | --- |
| ***Cas d'utilisation "s'authentifier"*** | |
| ***Titre*** | S'authentifier |
| ***Acteur*** | * Supervisor * Admin * Super Admin |
| ***Objectif*** | Permettre aux utilisateurs de se connecter afin de pouvoir accéder à leurs espaces de travail |
| ***Description des enchaînements*** | |
| ***Pré condition*** | ***Post condition*** |
| L’utilisateur doit avoir un compte (login et un mot de passe). | Quand la connexion est réussite, l’utilisateur est redirigé vers son tableau de bord. |
| ***Scénario nominal*** | |
| 1. L’utilisateur lance l’application. 2. Le système affiche la page d’accueil. 3. L’utilisateur clique sur le bouton «login». 4. Le système affiche la page d’authentification. 5. L’utilisateur saisie ses coordonnées. 6. Le système vérifie les coordonnées saisies. 7. L’utilisateur est redirigé vers sontableau de bord. | |
| ***Scénario alternatif*** | |
| A1 Les coordonnées saisies sont incorrectes  L'enchainement redémarre après le point 3 du scénario nominal  4. Un message d'erreur s'affiche sur l'écran  5. Le système va afficher en boucle la page d'authentification et revient au point 4 du scénario nominal jusqu'à ce que l'utilisateur réussisse l'authentification. | |

Tableau 10: Description textuelle de «s’authentifier»

##### 1.2.1.2.Le diagramme de séquence système

Dans le diagramme de séquence système nous modélisons ce dernier comme étant une "boîte noire" et nous nous concentrons essentiellement sur les interactions acteurs/système. Le fonctionnement d'un cas d'utilisation est notamment décrit sous la forme d’une séquencede messages échangés entre les acteurs et le système.

Le diagramme représenté dans la figure décrit le cas d’utilisation «s'authentifier»



Figure 11: Diagramme séquence système «s’authentifier»

#### 1.2.2.Conception de la vue dynamique:

Dans cette partie nous allons montrer le fonctionnement général de l'application par le biais d'un diagramme de séquence objet

##### 1.2.2.1.Diagramme de séquence objet

Dans cette partie nous citons les interactions acteur/système, tout en détaillant les actes du système d’un point de vue temporel

Figure 12: Diagramme séquence objet «s’authentifier»

#### 1.2.3.Réalisation

L’interface d’authentification permet à nos utilisateurs de se connecter à l’application



Figure 13: Première interface de l'application

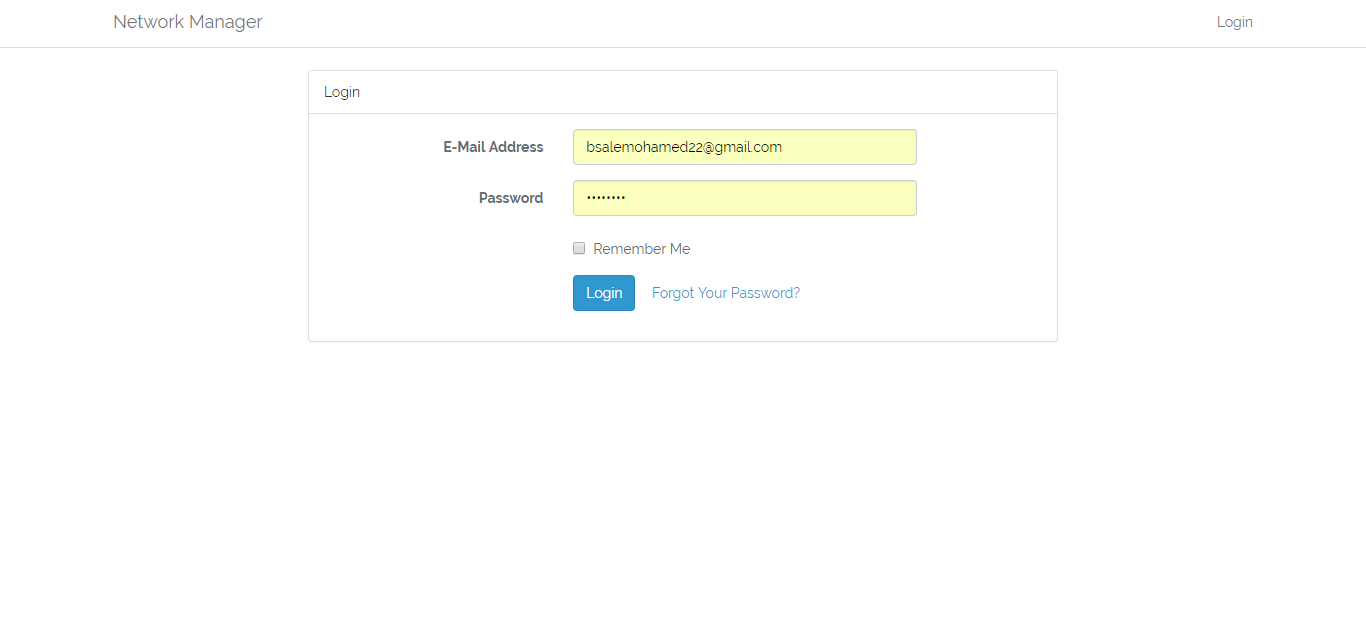


Figure 14: Interface de Login

#### 1.2.4. Tableau de tâches

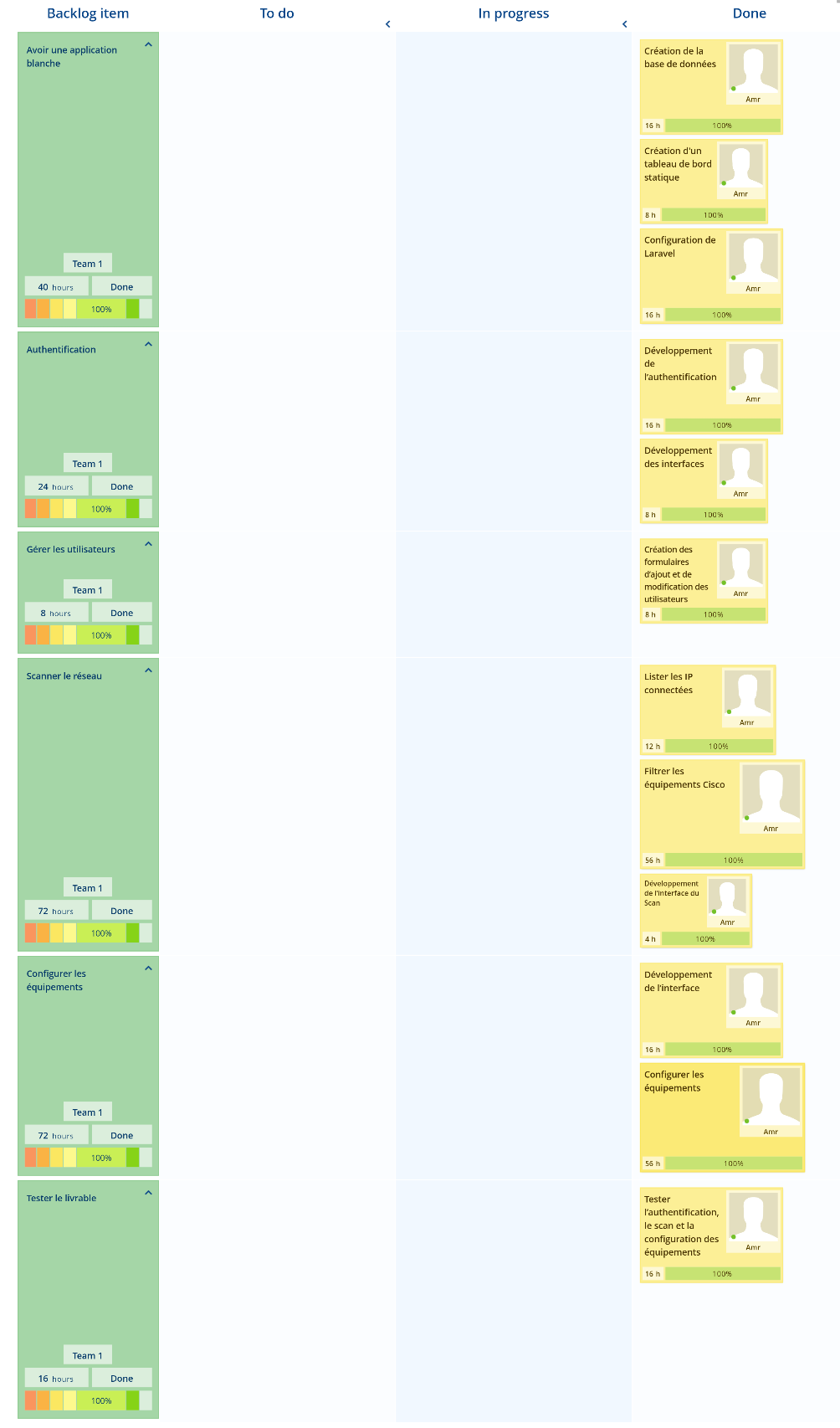


Figure 15: Tableau de tâches du Sprint 1

## 2.Sprint2:

### 2.1.Backlog du sprint 2

* Date de début du sprint : 2 Mars 2017
* Date de fin du sprint : 22 Mars 2017
* Temps estimé en heurs : 160h
* Echelle de mesure : Une journée de travail est égale à 8h de travail
* Objectif du sprint :
* Gérer les templates.
* Lister les équipements et les interfaces.
* Recherche avancée dans les listes.

|  |  |  |
| --- | --- | --- |
| ***Element du backlog*** | ***Tâches*** | ***Estimation*** |
| Avoir un système de templates | Création d’un système de templates | 56h |
| Gérer les templates | Création de l’interface de gestion des templates | 8h |
| Développement de la gestion des templates | 24h |
| Avoir une liste des équipements et leurs informations | Développement de l interface des équipements | 4h |
| Extraction et affichage des informations de chaque équipement | 20h |
| Rechercher dans la liste des équipements | Développement de la recherche dans la liste des équipements | 8h |
| Avoir la liste des interfaces de chaque équipement | Extraction et affichage des informations de chaque interface. | 24h |
| Tester le livrable | Tester le système de template et sa gestion, les listes des équipements et des interfaces. | 16h |

Tableau 11: Backlog du Sprint2

### 2.2.Vue technique

Dans cette partie, nous réalisons la spécification des besoins fonctionnels, la conception dynamique et les interfaces du second sprint.

#### 2.2.1.Spécification des besoins fonctionnels

Pour spécifier les besoins fonctionnels du second sprint, nous réalisons la description textuelle puis les diagrammes des séquences système.

##### 2.2.1.1.Description textuel de «Gestion des templates (Ajout)»

Le tableau suivant présente la description textuelle de «Gestion des templates (Ajout)»

|  |  |
| --- | --- |
| ***Cas d'utilisation "Gestion des templates (Ajout) "*** | |
| ***Titre*** | Ajout des templates |
| ***Acteur*** | * Admin * Super Admin |
| ***Objectif*** | Permettre aux utilisateurs d’ajouter un template. |
| ***Description des enchaînements*** | |
| ***Pré condition*** | ***Post condition*** |
| L’utilisateur doit être connecté | Quand l’ajout est réussi, le template est ajouté avec succès. |
| ***Scénario nominal*** | |
| 1. L’utilisateur choisi «Templates» depuis le tableau de bord. 2. Le système affiche la page demandée. 3. L’utilisateur remplit le formulaire. 4. L’utilisateur clique sur le bouton «Save». 5. Le système vérifie les données saisies. 6. Le système affiche un message de succès. | |
| ***Scénario alternatif*** | |
| A1 Le titre de template existe déjà/ Champ vide  L'enchainement redémarre après le point 3 du scénario nominal  6. Un message d'erreur s'affiche sur l'écran  7. Le système va afficher en boucle la page de templates et revient au point 3 du scénario nominal jusqu'à ce que l'utilisateur réussisse l'authentification. | |

Tableau 12: Description textuelle de «Gestion des templates (Ajout)»

##### 2.2.1.2.Le diagramme de séquence système

Le diagramme représenté dans la figure décrit le cas d’utilisation «Gestion des templates (Ajout)»



Figure 16: Diagramme séquence système «Gestion des templates (Ajout)»

#### 2.2.2.Conception de la vue dynamique:

Dans cette partie nous allons montrer le fonctionnement général de l'application par l biais d'un diagramme de séquence objet

##### 2.2.2.1.Diagramme de séquence objet

Dans cette partie nous citons les interactions acteur/système, tout en détaillons les actes du système d’un point de vue temporel

Figure 17: Diagramme séquence objet «Gestion des templates (Ajout)»

#### 2.2.3.Réalisation

La figure suivante présente les différentsinterfacesde l'application concernantla partie des templates.

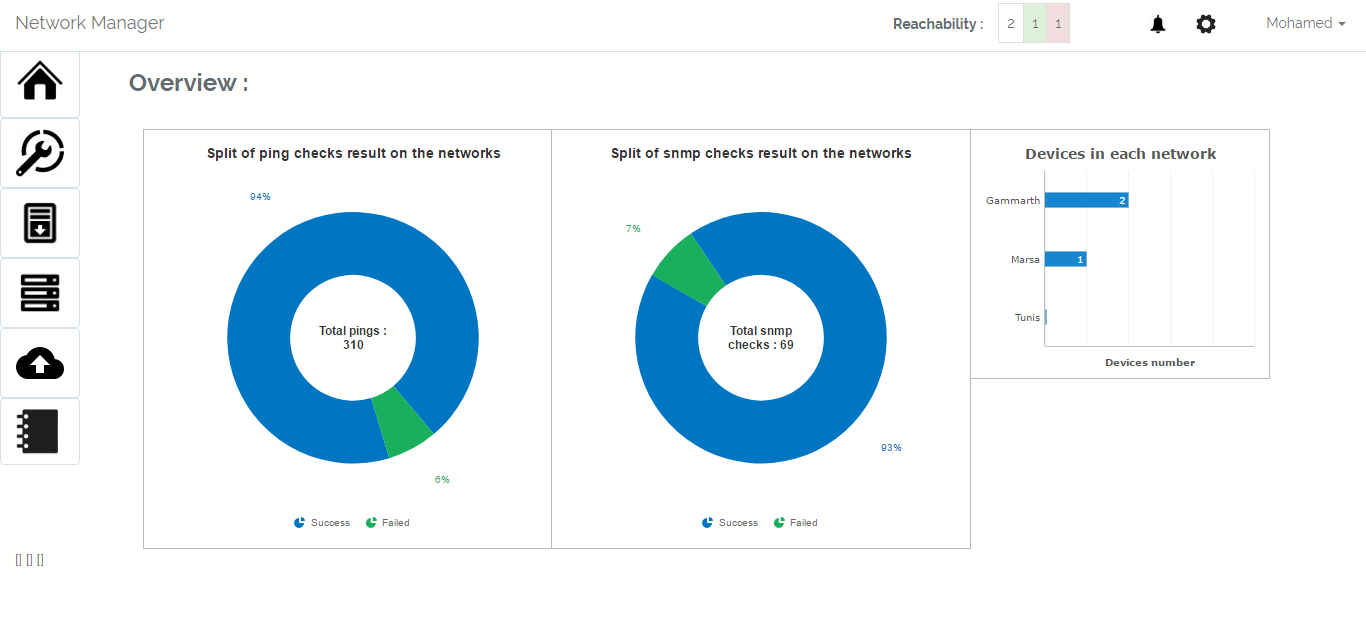


Figure 18: Interface d'accueil de l'application

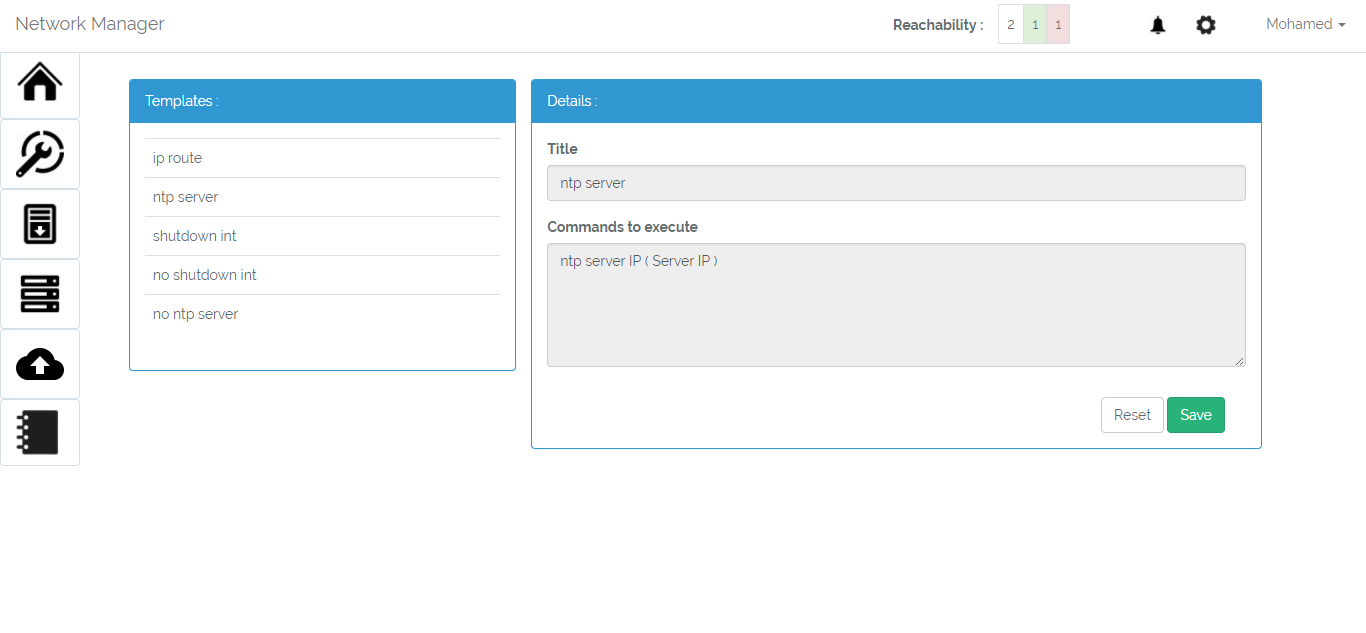


Figure 19: Interface de Templates

#### Sprint 22.2.4. Tableau de tâches

Figure 20: Tableau de tâches du Sprint 2

## 3.Sprint3

### 3.1.Backlog du sprint 3

* Date de début du sprint : 23 Mars 2017
* Date de fin du sprint : 03Avril 2017
* Temps estimé en heurs : 88h
* Echelle de mesure : Une journée de travail est égale à 8h de travail
* Objectif du sprint :
* Backup des configurations des équipements sur un serveur TFTP
* Restore des configurations des équipements depuis un serveur TFTP
* Ping Test sur les équipements

|  |  |  |
| --- | --- | --- |
| ***Element du backlog*** | ***Tâches*** | ***Estimation*** |
| Sauvegarder les configurations des équipements sur un serveur TFTP | Développement de l’interface | 4h |
| Développement de l’opération de backup | 20h |
| Consulter la configuration de chaque équipement | Développement de la consultation des configurations | 8h |
| Restaurer les dernières configurations sauvegardées | Mettre en place la dernière configuration sauvegardée dans startup config des équipements | 18h |
| Redemarrer les équipements | 6h |
| Gérer les réseaux | Ajout et suppression des réseaux | 16h |
| Tester PING | Développement du Ping Test | 8h |
| Tester le livrable | Tester le backup, la restauration, le ping et la gestion des réseaux | 8h |

Tableau 13: Backlog du Sprint3

### 3.2. Vue technique

#### 3.2.1.Spécification des besoins fonctionnels

##### 3.2.1.1.Description textuel de «Backup des configurations»

Avec l’aide de la description textuelle proposé par les concepteurs d’UML, les diagrammes aurons une meilleure lisibilité

Le tableau suivant présente la description textuelle de «Backup des configurations»

|  |  |
| --- | --- |
| ***Cas d'utilisation "Backup des configurations"*** | |
| ***Titre*** | Backup des configurations |
| ***Acteur*** | * Admin * Super Admin |
| ***Objectif*** | Permettre aux utilisateurs de sauvegarder les configurations des équipements. |
| ***Description des enchaînements*** | |
| ***Pré condition*** | ***Post condition*** |
| -L’utilisateur doit être connecté  -L’équipement doit être connecté | Quand la sauvegarde est faite, elle sera enregistrée dans la base de données |
| ***Scénario nominal*** | |
| 1. L’utilisateur choisit la rubrique « Backup ».  2. Le système affiche l’interface de backup.  3. L’utilisateur clique sur le bouton « Backup now ».  4. Le système affiche un message « Backup done successfully ». | |
| ***Scénario alternatif*** | |
| A1 Backup d’un seul équipement  L'enchainement redémarre après le point 2 du scénario nominal  3.L’utilisateur clique sur le bouton « + » du réseau.  4.Le système affiche les équipements du réseau.  5. L’utilisateur clique sur le bouton « Backup ».  6.Le système affiche un message « Backup done successfully ».  A2Echec du Backup  L'enchainement redémarre après le point 3 du scénario nominal  4. Le système affiche un message « Backup failed ». | |

Tableau 14: Description textuelle de «Backup des configurations »

##### 3.2.1.2.Le diagramme de séquence système

Le diagramme représenté dans la figure décrit le cas d’utilisation «Backup des configurations»



Figure 21: Diagramme séquence système «Backup des configurations»

#### 3.2.2.Conception de la vue dynamique

Dans cette partie nous allons montrer le fonctionnement général de l'application par l biais d'un diagramme de séquence objet

##### 3.2.2.1.Diagramme de séquence objet

La figure suivante représente le diagramme de séquence objet du cas d’utilisation «Backup des configurations»

Figure 22: Diagramme séquence objet « Backup des configurations»

#### 3.2.3.Réalisation

Les figures 24 et 25 présentent les différentes interfaces mentionnées ci dessus

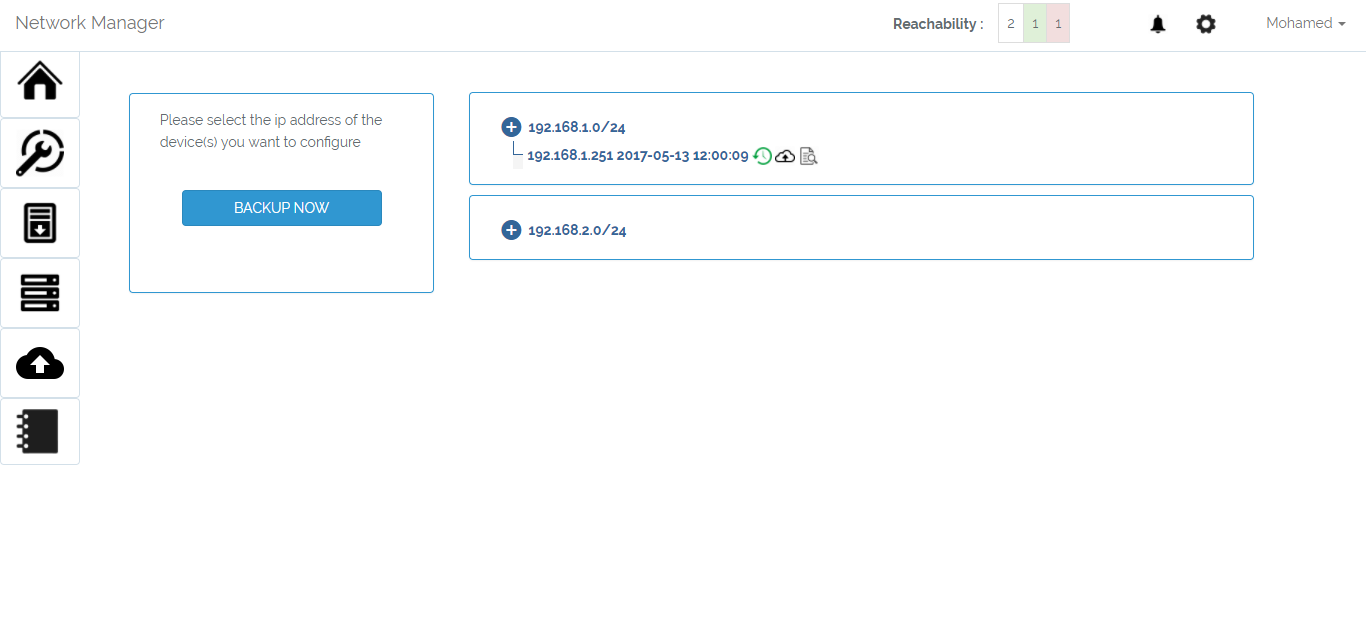


Figure 23:Interface du backups et restores

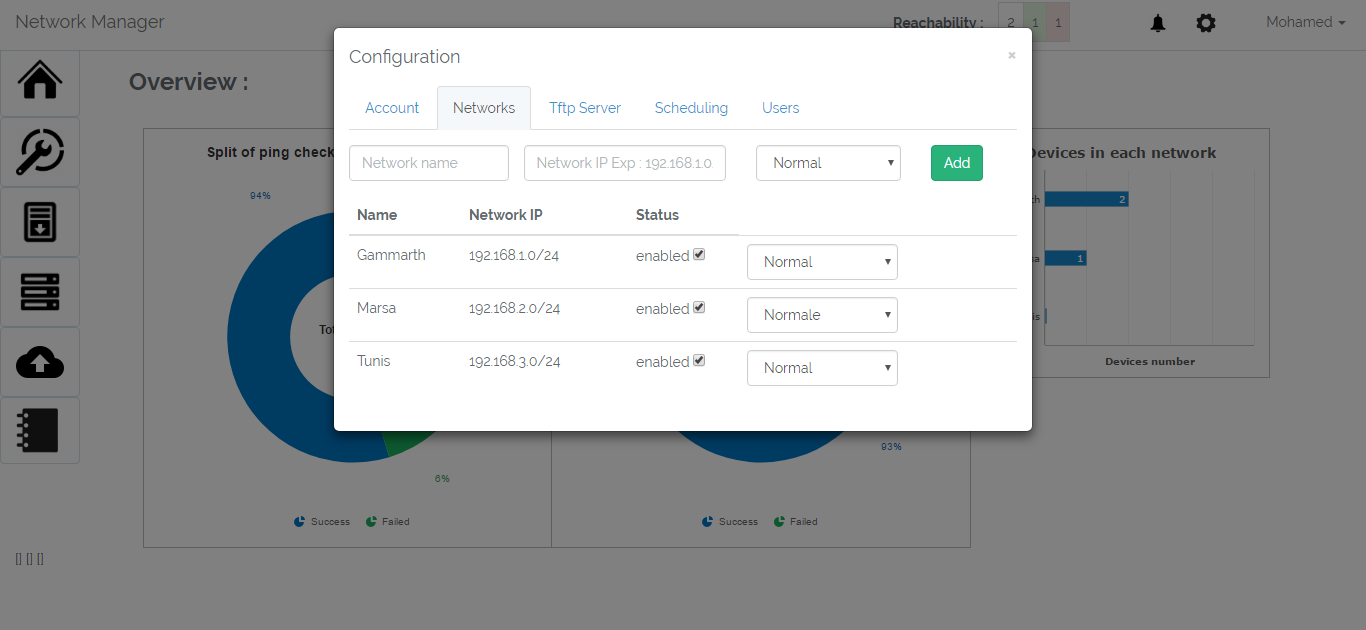


Figure 24; Interface de gestion des réseaux

## 4.Sprint4

### 4.1.Backlog du sprint 4

* Date de début du sprint : 04 Avril 2017.
* Date de fin du sprint : 13 mai 2017.
* Temps estimé en heurs : 320h.
* Echelle de mesure : Une journée de travail est égale à 8h de travail.
* Objectif du sprint :
* Programmation des tâches.
* Être notifié des informs.
* Consulter les statistiques, l’historique des informs et syslog.

|  |  |  |
| --- | --- | --- |
| ***Element du backlog*** | ***Tâches*** | ***Estimation*** |
| Programmer l’application à faire des tâches | Création de l’interface | 24h |
| Développementdes méthodes de programmations | 96h |
| Être notifié des informs SNMP | Développement de l’interface de notifications (pop-up) | 24h |
| Configuration du serveur | 90h |
| Développement des méthodes de notifications | 46h |
| Consulter les statistiques | Développement des méthodes de calculs des statistiques | 16h |
| Création des interfaces | 8h |
| Consulter l’historique des informs et syslog | Création de l’interface de l’historique | 8h |
| Rechercher un informs ou syslog dans l’historique | Développement de la méthode de recherche | 8h |

Tableau 15: Backlog du Sprint 4

### 4.2.Vue technique

#### 4.2.1.Spécification des besoins fonctionnels

##### 4.2.1.1.Description textuel de «Programmer l’application à faire des tâches»

Avec l’aide de la description textuelle proposé par les concepteurs d’UML, les diagrammes aurons une meilleure lisibilité

Le tableau suivant présente la description textuelle de «Programmer l’application à faire des taches programmées»

|  |  |
| --- | --- |
| ***Cas d'utilisation "Programmer l’application à faire des tâches"*** | |
| ***Titre*** | Programmer l’application à faire des tâches |
| ***Acteur*** | * Supervisor * Admin * Super Admin |
| ***Objectif*** | Programmer l’application à faire des taches. |
| ***Description des enchaînements*** | |
| ***Pré condition*** | ***Post condition*** |
| L’utilisateur doit être connecté | Quand l’ajout est réussi, les données sont enregistrées dans la base de données |
| ***Scénario nominal*** | |
| 1. L’utilisateur choisi «Templates» depuis le tableau de bord. 2. Le système affiche la page demandée. 3. L’utilisateur remplit le formulaire. 4. L’utilisateur clique sur le bouton «Save». 5. Le système vérifie les données saisies. 6. Le système affiche un message de succès. | |
| ***Scénario alternatif*** | |
| A1 Le titre de template existe déjà / Champ vide  L'enchainement redémarre après le point 3 du scénario nominal  6. Un message d'erreur s'affiche sur l'écran  7. Le système va afficher en boucle la page de templates et revient au point 3 du scénario nominal jusqu'à ce que l'utilisateur réussisse l'authentification. | |

Tableau 16: Description textuelle de «Programmer l’application à faire des tâches»

##### 4.2.1.2.Le diagramme de séquence système

Le diagramme représenté dans la figure décrit le cas d’utilisation «Programmer l’application à faire des tâches»



Figure 25: Diagramme séquence système «Programmer l’application à faire des taches programmées»

#### 4.2.2.Conception de la vue dynamique

Dans cette partie nous allons montrer le fonctionnement général de l'application par l biais d'un diagramme de séquence objet

##### 4.2.2.1.Diagramme de séquence objet

Dans cette partie nous citons les interactions acteur/système, tout en détaillons les actes du système d’un point de vue temporel



Figure 26: Diagramme séquence objet « Programmer l’application à faire des taches programmées »

#### 4.2.3.Réalisation

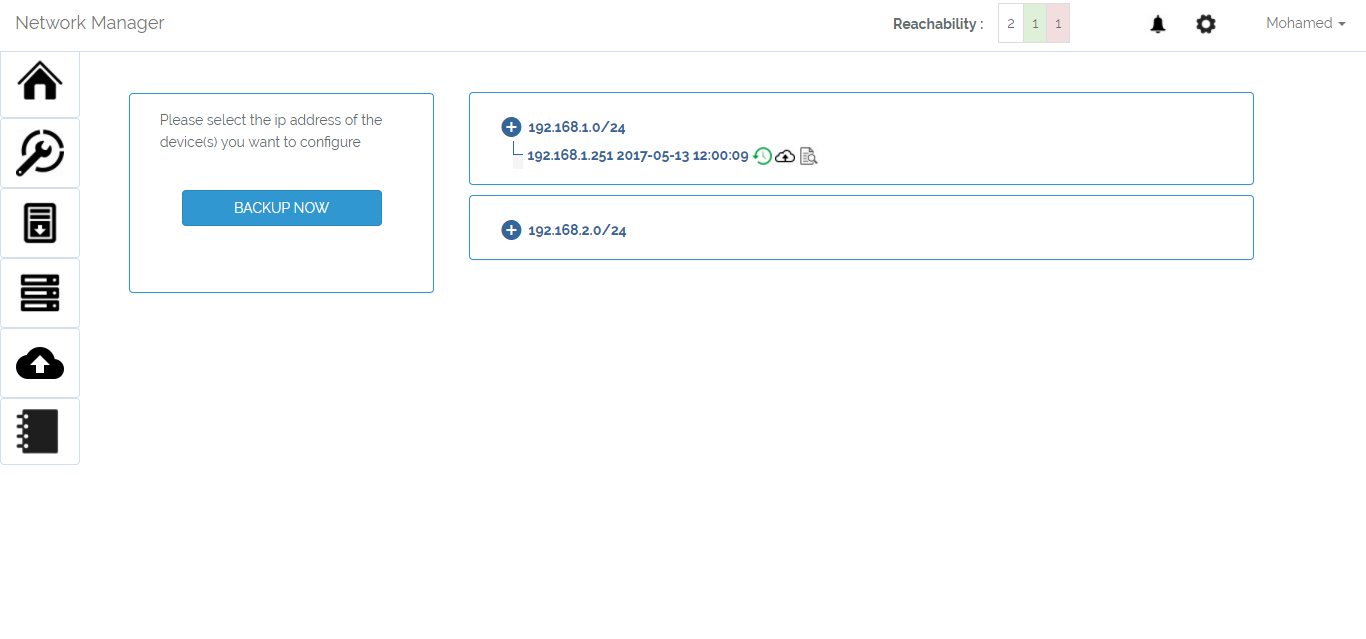


Figure 27: Interface de Backups

## 5.Maquette

Avant de concevoir une page de point de vue couleurs, graphismes ou animations, il faut concevoir un gabarit pour les pages principales de l’application qui seront par la suite héritées par les autres pages.

### 5.1.Gabarit de la page d’authentification

Lors de lancement de l’application, une interface d’authentification s'affiche contenant un formulaire d’indentification. La (Figure) illustre le gabarit de cette interface.

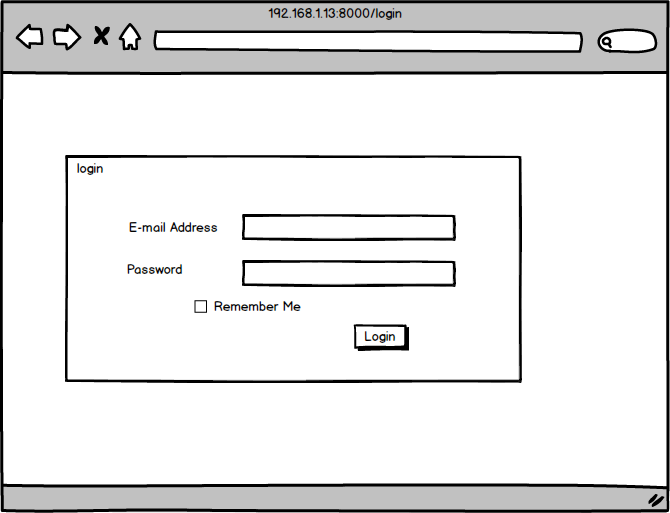


Figure 28: Gabrit de la page « d’authentification »

### 5.2.Gabarits des pages de l’application

La figure .montre la structure générale des interfaces de notre application et la disposition de ces différentes parties.

Leheaderen haut de la page contient le nom de l'utilisateur connecté, le bouton de déconnexion, paramètres, notifications, et nom de l’application.

A gauche nous trouvons la barre de navigation entre les différentes interfacesde l’application.

La zone principale qui est la plus grande, elle affiche l'interface relative au panneau sélectionné.

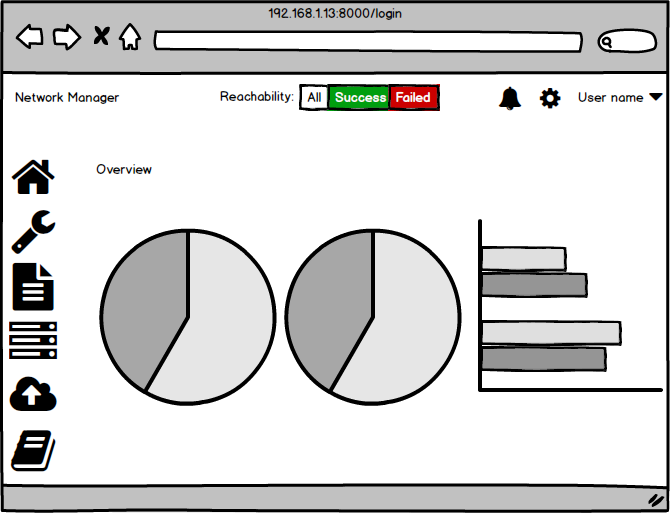


Figure 29: Gabarit global de l’application

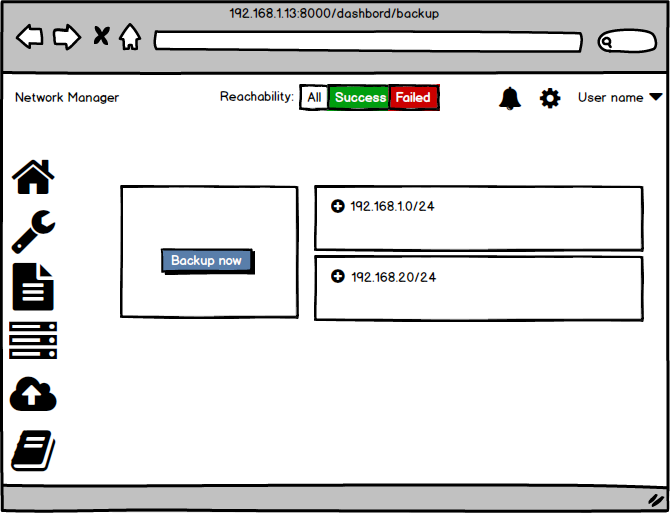


Figure 30: Gabarit de la configuration des équipements

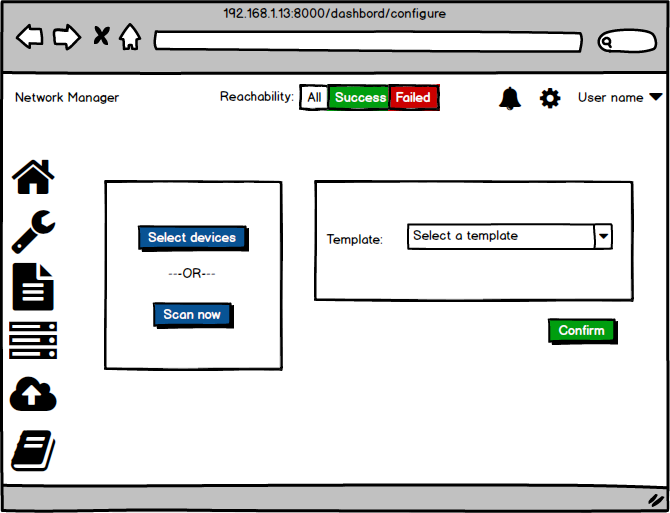


Figure 31: Gabarit du backup et du restore de l’application

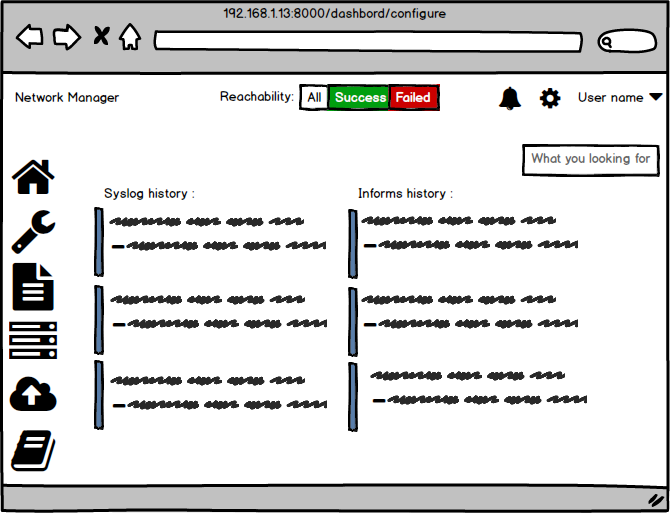


Figure 32: Gabarit de la rubrique historique des informs et syslog

## 6. Burndown chart des quatre Sprints

La figure suivante illustre les burndown chart des quatre sprints de ce release. chaque burndown chart est réalisé à la fin d’un sprint grâce à des backlog détaillés. Ces représentations graphiques décrivent l’avancement du travail par rapport au temps estimé.

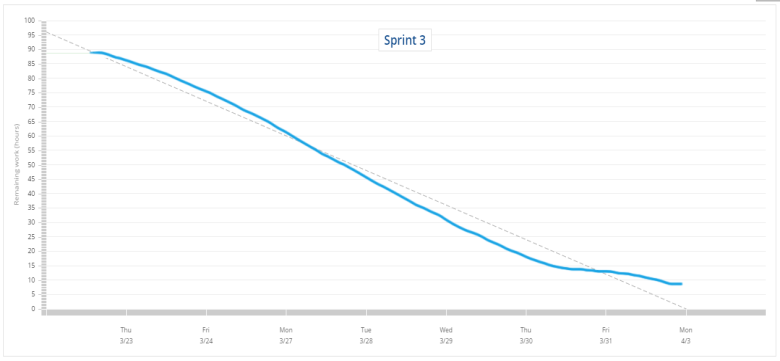
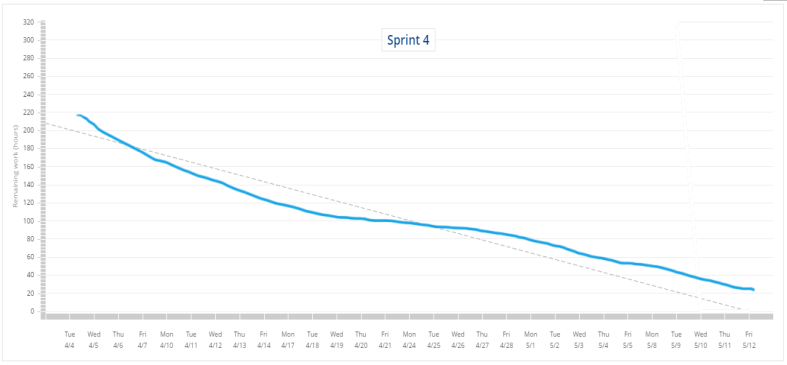
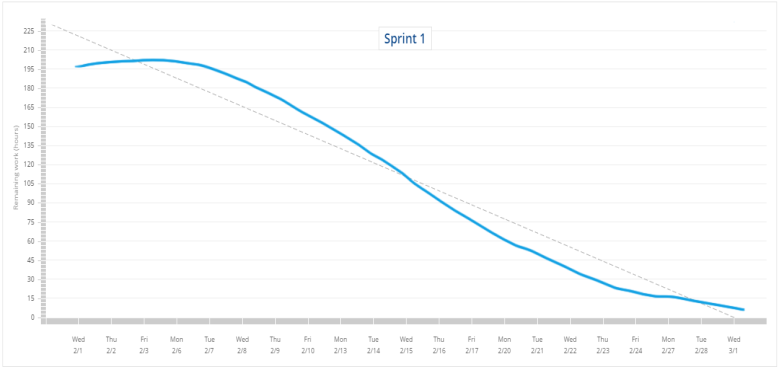
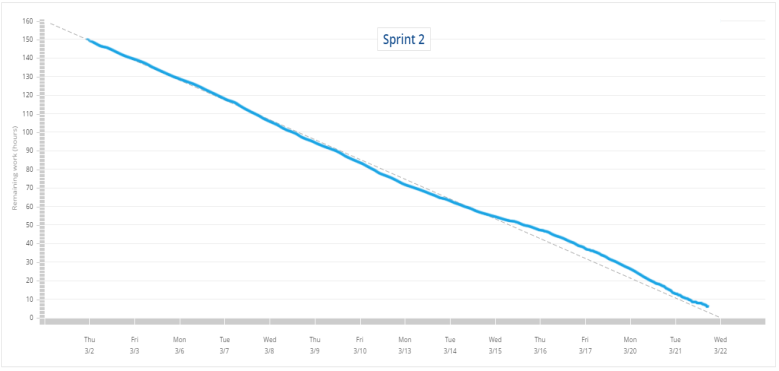


Figure 33: Burndown chart

## Conclusion

A la fin de ce chapitre, notre application est finie et le release est achevé. Un utilisateur peut effectuer toutes les fonctionnalités. Durant ce chapitre nous avons présenté la vue graphique et technique de chaque sprint ainsi que la partie réalisation. Maintenant il ne nous reste que tester notre application afin de mesurer le degré de satisfaction des utilisateurs, c’est la dernière partie de notre projet.

# Test et validation

L

e chapitre présent nommé «Test et validation» est le dernier chapitre de notre rapport.

Danscettepartienousvériﬁonssilesbesoinsfonctionnelsetnonfonctionnelsconcernantleproduitetlaméthodologiespéciﬁésaudébutduprojetsontrespectésoupasgrâceaumoyen d’évaluation.

## Évaluation du produit

L’évaluation précise du produit s’intéresse à la validation de la qualité à la production et à l’utilisation.Danslapremièrepartiel’évaluationdelaqualitéàlaproduction,nousexpliquons les différent tests effectués puis dans la deuxième section nous décrivons le test et l’évaluation de l’application par les utilisateurs ﬁnaux.

### 1.1. Architecture réseaux

Pour pouvoir tester notre application, nous avons conçu un prototype minimaliste de l’architecture réseau WAN et LAN de la BIAT. Ce prototype se compose d'un Vlan du siège comportant notre serveur de supervision et de configuration, d'un réseau WAN reliant les CE ( customer edge ) de la Banque en Utilisant l'infrastructure d'un fournisseur de services et des réseaux de quelque agences.

### GNSFINAL

Figure 34:Architecture réseau

**Configuration des équipements :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Equipement | Interface | Adresse IP | Masque de sous réseau | Routage | Réseau |
| Siège | F0/0 | 192.168.3.1 | 255.255.255.0 | OSPF | 192.168.3.0 |
| F0/1 | 192.168.1.254 | 255.255.255.0 | OSPF | 192.168.1.0 |
| MarsaR2 | F0/0 | 192.168.3.2 | 255.255.255.0 | OSPF | 192.168.3.0 |
| F0/1 | 192.168.2.1 | 255.255.255.0 | OSPF | 192.168.2.0 |
| TunisR3 | F0/0 | 192.168.3.3 | 255.255.255.0 | OSPF | 192.168.3.0 |
| F0/1 | 192.168.4.1 | 255.255.255.0 | OSPF | 192.168.4.0 |
| GabesR4 | F0/0 | 192.168.3.4 | 255.255.255.0 | OSPF | 192.168.3.0 |
| F0/1 | 192.168.5.1 | 255.255.255.0 | OSPF | 192.168.5.0 |
| GafsaR5 | F0/0 | 192.168.3.5 | 255.255.255.0 | OSPF | 192.168.3.0 |
| F0/1 | 192.168.6.1 | 255.255.255.0 | OSPF | 192.168.6.0 |

Tableau 17: Configuration des équipements

### 1.2.Scénario de test

Le scénario de test ci-dessous permet de décrire la procédure d’utilisation de l’application après sa mise en place.

*L’administrateur se connecte à l’application à travers l’interface de login, si son login et son mot de passe sont valides, il sera redirigé vers le tableau de bord , sinon il reçoit un message d’erreur. Une fois au tableau de bord , il clique sur le bouton de paramètres (Settings) ensuite sur utilisateurs (Users). Il remplit le formulaire afin de créer un utilisateur.*

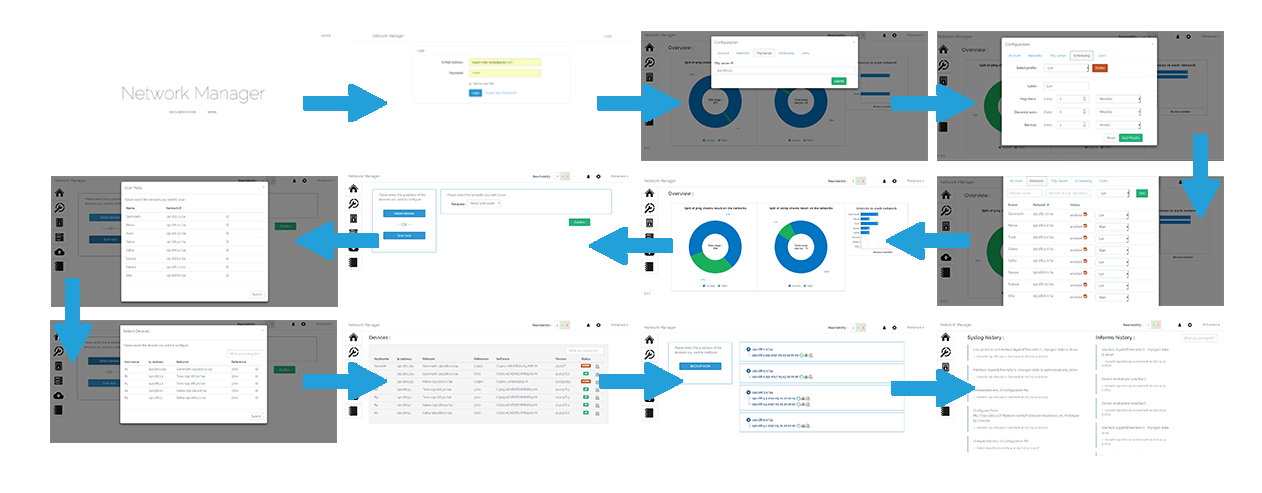
*L’utilisateur se connecte à l’application à travers l’interface de login, si son login et son mot de passe sont valides, il sera redirigé vers le tableau de bord , sinon il reçoit un message d’erreur. Une fois au tableau de bord,il clique sur le bouton de paramètres (Settings) et choisi l’onglet Tftp et saisi l’adresse IP du serveur TFTP puis valide. Ensuite il clique sur l’onglet « Scheduling » afin de créer un profil de programmation de scan des réseaux , il remplit le formulaire et valide. Toujours dans les paramètres (Settings), l’utilisateur choisi l’onglet « Networks » et remplit le formulaire puis choisi le profil adéquat pour chaque réseau puis valide d’ajouter un réseau. L’utilisateur choisit la rubrique « Templates », remplit le formulaire et valide pour ajouter un Template.Pour configurer des équipements, l’utilisateur clique sur la rubrique « Configure » il lance une découverte de réseaux instantané en cliquant sur le bouton « Scan now » et en choisissant les réseaux à scanner. Un écran de chargement apparaît puis disparait quand la découverte est finie. L’utilisateur clique sur « Select Devices » pour cocher les équipements concernés ,il choisitun « Template » puis remplit le formulaire de ce dernier et clique sur « Confirm ». Le système ensuite affiche un message de succès ou un message d’erreur s’il y a une erreur. Pour faire une sauvegarde (ou une restauration) des configurations des équipements, l’utilisateur choisi la rubrique « Backups » puis clique soit sur « Backup now » pour faire un ‘backup’ sur tous les équipements connectés , soit sur le bouton de backup d’un seul équipement pour faire un ‘backup’ d’un seul équipement.. Pour consulter tous les équipements enregistrés, l’utilisateur choisi la rubrique « Devices ». Pour consulter l’historique des informs et syslogs (Notifications), il choisi la rubrique « Logs ».*

### 1.3.Validation de la qualité à la production

Aﬁn de contrôler le respect des conventions de codage présentées dans la partie «besoins non fonctionnels internes», nous avons effectué des tests de fonctionnement de l’application tout au long du cycle de développement.

#### 1.3.1.Tests de l’application

Nous avons testé les différentes fonctionnalités de l’application en suivant le scénario de test décrit dans le paragraphe précédent.

Figure 35 Le déroulement du test

#### 1.3.2.Test de la qualité à la production logicielle

L’évaluation interne du produit consiste à tester les caractéristiques du logiciel tel que la sécurité.

##### 1.3.2.1.Évaluation de la sécurité

Vu que la sécurité est un critère essentiel pour NetworkManager, nous avons analysé l’application avec l’outil YASCA.

###### 1.3.2.1.1.Présentation de l’outil YASCA

Yasca est un outil open source qui fait le scan des failles de sécurité, de la qualité, de la performance et de la conformité aux meilleures pratiques dans le code source d’un programme. Il s’agit d’un outil en ligne de commande qui génère des rapports au format HTML, CSV, XML, SQLite et d’autres formats.(N12)

1.3.2.1.2.Résultat de test

Aprèsl’analysedelasécuritéducode,YASCAafﬁcheunrapportcontenantleserreursqu’on peut corriger pour avoir un code plus sécurisé.

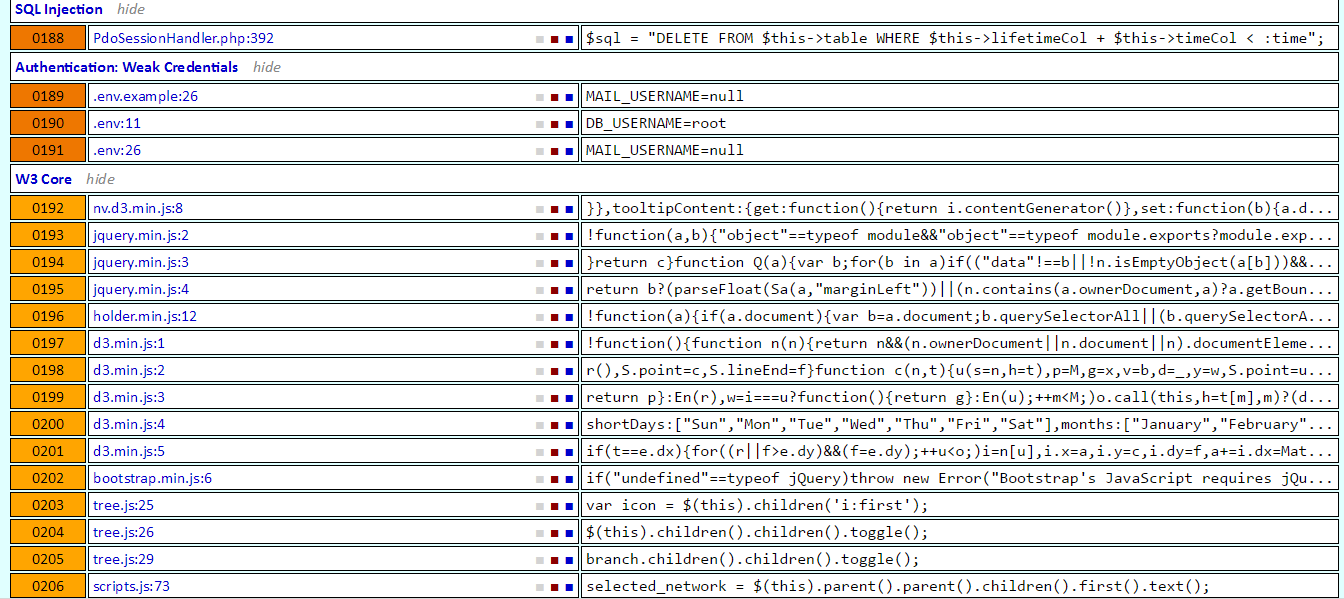
La ﬁgure suivante illustre le résultat de test effectué avec YASCA

Figure 36 : Résultat du test

### 1.4. Validation à l'utilisation

Durant l’étape de validation de la qualité à l’utilisation, nous avons utilisé le questionnaire comme outil d’évaluation principale. Nous avons proposé ce questionnaire à des utilisateurs finaux après avoir testé l'application.

#### 1.4.1. Procédure de test

Pour évaluer la qualité à l'utilisation, nous avons réalisé les étapes suivantes tout en se basant sur le scénario de test mentionné ci-dessus.

**1ère étape:** exécution du scénario de test par les utilisateurs suscités ;

**2ème étape:** distribution du questionnaire ;

**3ème étape:** récupération des réponses ;

**4ème étape:** génération des statistiques ;

**5ème étape:** analyse des résultats du questionnaire;

#### 1.4.2. Questionnaire de test de validation

Le test utilisateur vise à évaluer notre application et non l'utilisateur. Cette démarche exige une approche professionnelle et une méthodologie précise. Il est donc primordial de préparer un questionnaire en se basant sur les exigences de qualité mentionnées ci-dessus.

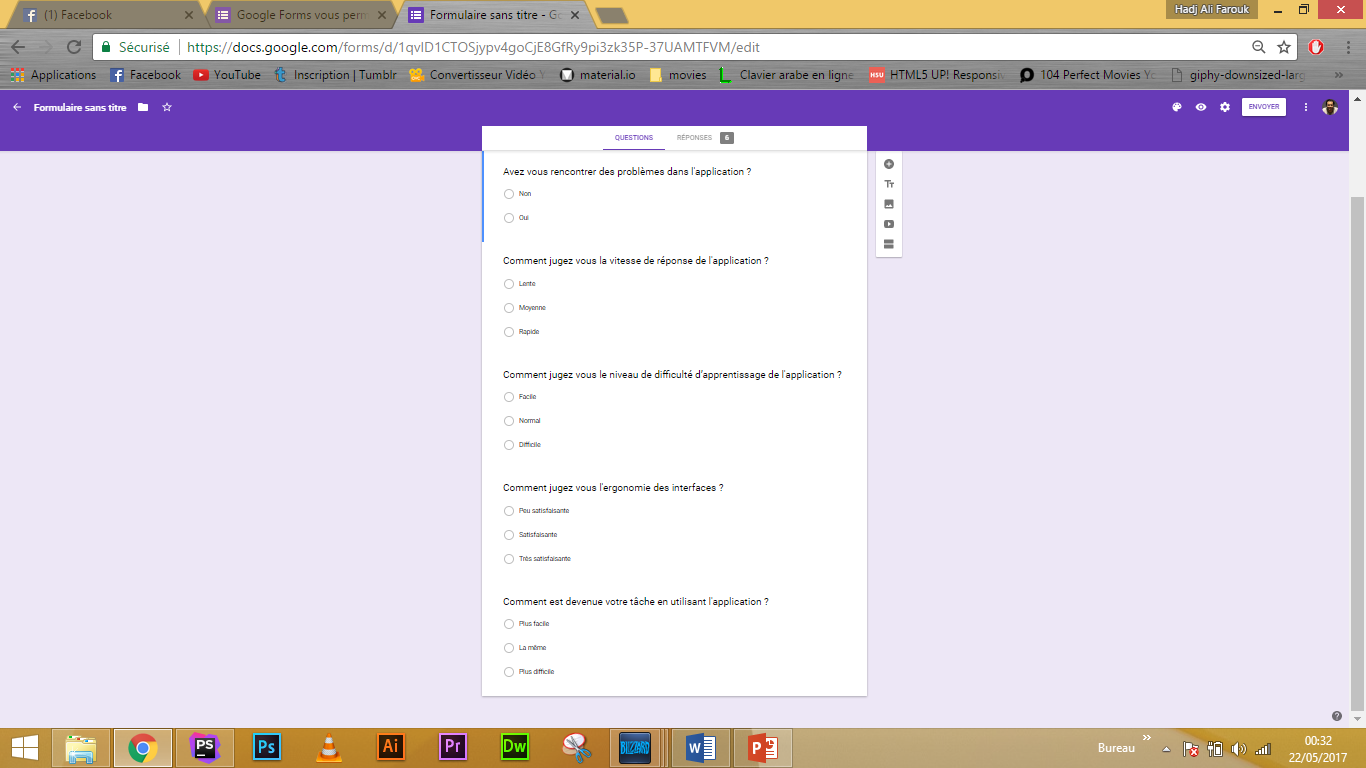


Figure 37: Questionnaire de satisfaction

#### 1.4.3. Evaluation de la satisfaction

Grâce à l’exécution du scénario de test et au questionnaire, nous avons évalué la qualité du projet. Les résultats du questionnaire et leurs analyses sont représentés dans la figure suivante

### 65

### 321

Figure 38: Evaluation de la satisfaction

### 

### 1.4. Analyse de l’efficacité

L'efficacité et l'utilisabilité d'une application sont deux termes reliés. La norme ISO 9241-11 définit l’utilisabilité de la manière suivante : « Un système est utilisable lorsqu’il permet à l’utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction dans le contexte d’utilisation spécifié ».les tâches, quelques-uns ont demandé une certaine assistance pour terminer ces tâches. Au niveau de l’authentification, les utilisateurs nous ont demandé s’il y avait une méthode de sécurité au niveau de la saisie de leurs mots de passe, nous les avons renseigné qu’il y a un hachage automatique généré par la Framework Laravel.

## Conclusion

Durant la phase de test, nous avons détecté quelque erreurs et mauvaise pratiques de codage que nous avons corrigé pour avoir un produit professionnel et de qualité.

# Netographie

N1. BIAT: http://www.tunisieindex.com/entreprises/Banque-BIAT-TUNISIE-48.html

N2. BIAT: http://www.etudier.com/dissertations/Timo/463704.html

N3.

N4.

N5. Nagios: https://sites.google.com/site/plecorentin/portefeuille-de-competence/travaux-menes-au-cfai84/serveur

N6. Scrum: www.agiliste.fr/introduction-methodes-agiles/

N7. eXtreme Programming: http://medina.developpez.com/cours/extreme-programming/

N8. UML: https://fr.wikipedia.org/wiki/UML\_(informatique)

N9.

N10. MVC: https://fr.wikipedia.org/wiki/Modèle-vue-contrôleur

N11. Laravel: https://laravel.com/docs/5.4