

# *Plan de développement*

## Nouvelle interface web OpenWrt

Les informations d'identification du document et les éléments de vérification du document

Référence du document :	P1	Validé par :	Baptiste Jonglez
Version du document :	2.0	Validé le :	08/01/2020
Date du document :	08/01/2020	Soumis le :	08/01/2020
Auteur(s) :	ZHANG Zhao	Type de diffusion :	Document électronique (.pdf)
Mots clés : OpenWrt、LuCI、planification du développement、tâche détaillée、exemple de fonctionnalité			

## Sommaire

1. Introduction	2
1.1 Objectifs et méthodes	2
1.2 Documents de référence	2
2. Concepts de base	2
3. Organisation	3
3.1 Décomposition en tâches	3
3.2 Ressources Humaines nécessaires	4
4. Planification	5
5. Méthodes et outils	5
6. Fonctionnalités	6
6.1 Configuration le nom (SSID) et le mot de passe du WIFI	6
6.2 Consultation l'état de connexion du réseau (IP publique, DNS)	7
7. Qualité	8
8. Glossaire	8
9. Références	8

# 1. Introduction

L'objet du document vise à faciliter l'intégration d'interface web à OpenWrt par les développeurs et vise à garantir que l'interface web correspondra bien aux besoins de l'utilisateur final.

Ce document contient toutes les références de documentation générale et technique ainsi que l'organisation d'équipe, la planification du développement, les tâches détaillées et les fonctionnalités définies.

## 1.1 Objectifs et méthodes

L'objet est de créer une nouvelle interface web rapide et efficace qui se base sur LuCI en développant HTML, CSS et JavaScript. et puis, le code doit être propre, logique et extensible.

Ce document permet d'aider le développeur à comprendre les besoins d'utilisateur et les fonctionnalités à réaliser sans difficulté comme un exemple.

Le développement sera fait dans le dépôt git de LuCI sur Gitlab et se basera l'environnement sur Docker.

## 1.2 Documents de référence

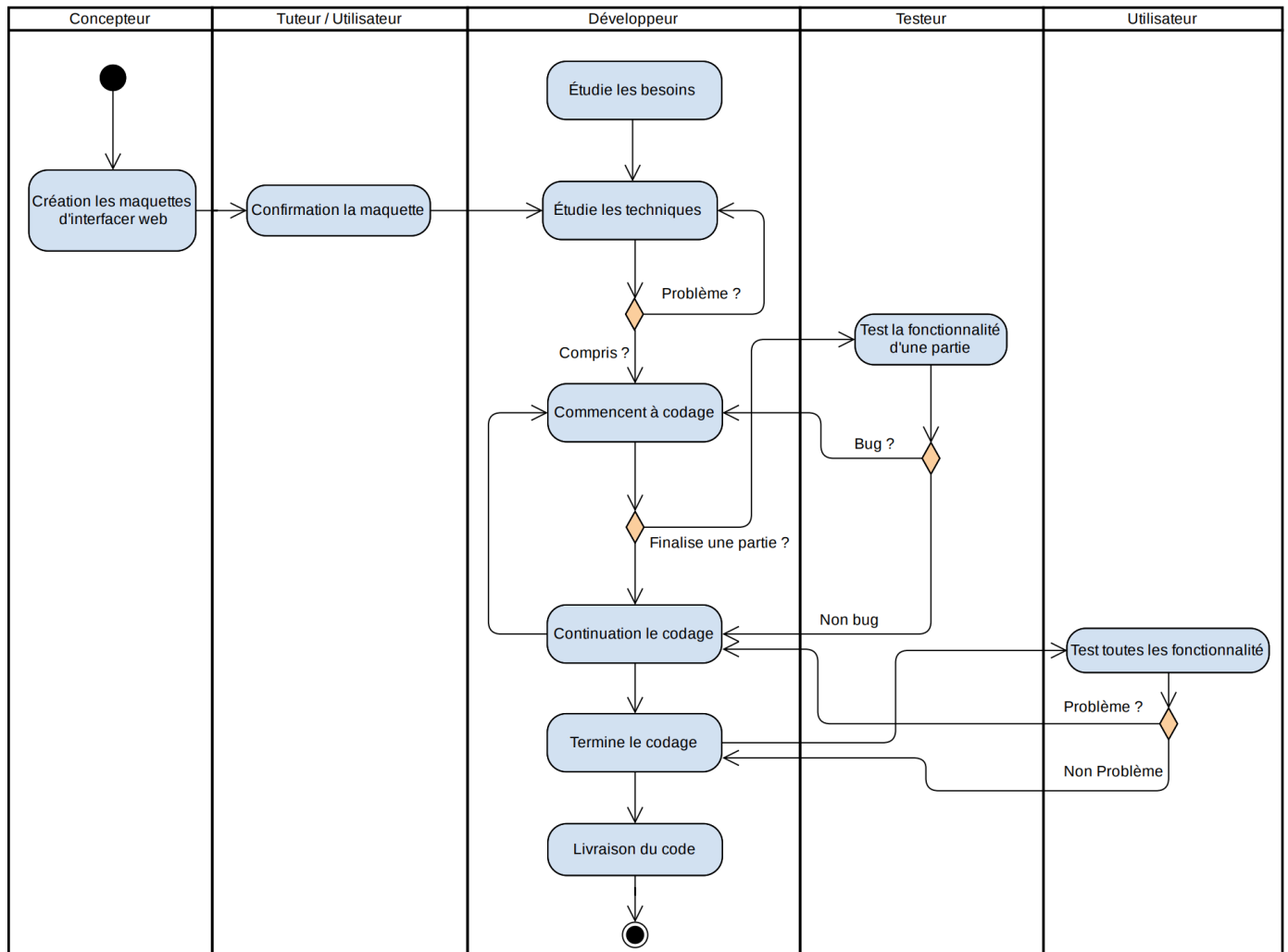
- Le cahiers des charges
- Le cahiers des recettes

# 2. Concepts de base

Il faut au moins connaître les bases du réseau et des diagrammes UML pour comprendre ce document.

Le développeur doit avoir les techniques fondamentaux du web (API, HTML, CSS et JavaScript)

## 3. Organisation



### 3.1 Décomposition en tâches

#### 3.1.1 Définir la conception d'interface web

- 1)、Trouver des idées selon le cahier des charges
- 2)、Faire des maquettes
- 3)、Confirmer une maquette finale

#### 3.1.2 Mise à niveau sur les techniques utilisés

- 1)、Apprendre les techniques de base du web (HTML、CSS、JavaScript)
- 2)、Étudier l'utilisation du LuCI-JS
- 3)、Lancer un environnement de développement en Docker
- 4)、Essayer de construire l'interface web avec la documentation de LuCI

### **3.1.3 Développement**

- 1)、 Découper les tâches du développement
- 2)、 Commencer à coder l'interface web
- 3)、 Tester chaque partie du développement réalisé

### **3.1.3 Faire un point de réunion du coté développeur**

- 1)、 Faire un point sur l'avancement du développement
- 2)、 Identifier les difficultés
- 3)、 Trouver les solutions ensemble

### **3.1.4 Tester les fonctionnalités et l'apparence sur les différents appareils**

- 1)、 Parcourir toutes les pages web pour identifier les problèmes d'affichage
- 2)、 Parcourir toutes les pages web pour identifier les problèmes de fonctionnalité.
- 3)、 Tester l'interface web sur un vrai routeur

### **3.1.5 Livraison l'interface web**

- 1)、 Valider le projet à partir du cahier de recettes
- 2)、 Interagir avec les développeurs de LuCI pour modifier et intégrer le code à LuCI

## **3.2 Ressources Humaines nécessaires**

- 1 concepteur de maquette pour l'interface
- 2 développeur frontend pour le codage
- 1 développeur backend pour nous aider du côté backend.
- 1 testeur pour tester chaque partie du développement
- 1 testeur pour tester le projet dans son ensemble

## 4. Planification

Tâche	Durée	Début	Fin
1) Créer une maquette finale (pc/mobile)	7 jours	13/01/2020	20/01/2020
2) Découper les tâches du développement	1 jours	21/01/2020	21/01/2020
3) Mise à niveau sur les techniques utilisés	31 jours	22/01/2020	21/02/2020
4) Commencer au codage du développement	67 jours	24/02/2020	30/04/2020
5) Faire un point de réunion du coté développeur	67 jours	24/02/2020	30/04/2020
6) Tester chaque partie du développement réalisé	67 jours	24/02/2020	30/04/2020
7) Tester les fonctionnalités et l'apparence sur les différents appareils	43 jours	01/05/2020	12/06/2020
8) Corriger tous les bugs trouvés	43 jours	01/05/2020	12/06/2020
9) Livraison l'interface web	indéfinie	26/06/2020	indéfinie

## 5. Méthodes et outils

- UML : Faire les diagramme UML pour mieux comprendre le projet
- Docker : Simplifier l'environnement du développement aux développeurs
- Adobe illustre : Faire les maquettes
- Linux Serveur (Ex:AWS) : Mise en place l'interface web pour le test en publique
- Navigateur (Ex: Chrome) : Un navigateur à aider pour développer l'interface
- IDE Code (Ex: PhpStorm、 Sublime Text) : Un outil du codage

# 6. Fonctionnalités

## 6.1 Configuration le nom (SSID) et le mot de passe du WIFI

Exemple: Configuration le nom (SSID) et le mot de passe du WIFI en méthode de cryptage psk2

```
var wifi = new LuCI.Network.WifiNetwork().getNetwork();
wifi.set('SSID', 'nouveau nom');
wifi.set('encryption', 'psk2');
wifi.set('key', 'nouveau mot de passe');
```

Fonction	Type	Résultat	Description
new LuCI.Network.WifiNetwork().getNetwork()	Instance	null   LuCI.Network.Protocol	Renvoie une instance Network.Protocol représentant l'interface logique ou null si ce réseau n'est attaché à aucune interface logique.
set(opt, val)	Method		Définissez l'option UCI donnée de ce réseau à la valeur donnée.

Paramètre	Type	Description
opt	string	Nom de l'option UCI à définir.
val	null   string   Array.<string>	La valeur à définir ou null pour supprimer l'option donnée de la configuration.

## 6.2 Consultation l'état de connexion du réseau (IP publique, DNS)

Exemple: Consultation l'état de connexion du réseau (IP publique, DNS)

```
var device = new LuCI.Network.Protocol().getDevice();
var IPv4 = device.getIPAddr();
var dnsIPv4 = device.getDNSAddrs();
```

Fonction	Type	Résultat	Description
new LuCI.Network.Protocol().getDevice()	Instance	null   Array.<LuCI.Network.Device>	Renvoie une instance de classe Network.Device représentant le périphérique réseau Linux attendu en fonction de la configuration.
getIPAddr()	Method	null   string	Renvoie l'adresse IPv4 principale enregistrée par le gestionnaire de protocole ou null si aucune adresse IPv4 n'a été définie.
getDNSAddrs()	Method	null   string	Renvoie un tableau de serveurs DNS IPv4 enregistrés par le backend de protocole distant.

## 7. Qualité

- Efficacité: l'interface web doit fonctionner rapidement avec peu de mémoire du routeur.
- Adaptabilité : le responsive de l'interface web doit fonctionner bien avec les différentes tailles d'écran
- Maniabilité: l'autre développeur ne doit pas avoir trop de difficulté sur la partie de compréhension du code
- Testabilité: l'interface doit être facile à tester sur un vrai routeur ou un environnement Docker

## 8. Glossaire

- OpenWrt (OPEN Wireless Router): un projet open source pour système d'exploitation embarqué basé sur Linux, principalement utilisé sur les appareils embarqués avec l'interface Web (LuCI).
- Tous les composants ont été optimisés pour être suffisamment petits pour tenir dans le stockage et la mémoire limités disponibles dans les routeurs domestiques.
- LuCI: une interface web gratuite de configuration d'un routeur OpenWrt et elle se développe en framework MVC qui utilise le langage de programmation Lua et divise l'interface en parties logiques comme les modèles et les vues.
- Docker : un logiciel libre permettant facilement d'initialiser l'environnement du développement.
- JSON: un format de données textuelles dérivé de la notation des objets du langage JavaScript.
- JSON-RPC : est un protocole d'appel de procédure à distance codé en JSON.

## 9. Références

- HTML: <https://www.w3schools.com/html/>
- CSS: <https://www.w3schools.com/css/>
- JavaScript: <https://www.w3schools.com/js/>
- JSON-RPC : <https://www.jsonrpc.org/specification>
- Docker: <https://gricad-gitlab.univ-grenoble-alpes.fr/projet-ter-m1-wic-openwrt/luci/blob/docker/README.md>
- Doc OpenWrt: <https://openwrt.org/docs/guide-user/start>
- LuCI : <https://openwrt.org/docs/guide-user/LuCI/webinterface.overview>
- LuCI API : <http://openwrt.github.io/luci/jsapi/>
- LuCI Template: <https://github.com/openwrt/luci/wiki/Templates>
- OUI : <https://github.com/zhaojh329/oui/>
- JUCI : <https://github.com/mkschreder/juci>