CoronaAlert

Version 2.0



Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| <17/05/2020> | 1.0 | Création du document | Taha Lekhlaikh |
| <13/06/2020> | 2.0 | La deuxième version du document | Taha Lekhlaikh |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

2. Objectif du logiciel 4

2.1 Contexte 4

2.2 Besoins fonctionnels 4

2.3 Besoins non fonctionnels 4

3. Structure 4

3.1 Vue des couches 4

3.2 Sous-systèmes et paquetages 4

3.3 Interfaces 5

4. Comportement 5

4.1 Réalisation des cas d’utilisation 5

4.2 Mécanismes 5

5. Autres vues 5

5.1 Vue processus (optionnel) 5

5.2 Vue implémentation (optionnel) 5

5.3 Vue déploiement (optionnel) 6

5.4 Vue données (optionnel) 6

6. Concepts du domaine 6

7. Qualités de l’architecture 6

8. Points ouverts 6

Document d'Architecture du Logiciel

# Introduction

Le but du document est de décrire l’architecture du logiciel de notre projet CoronaAlerte concernant les différents modules qu’englobe ce dernier. La structure suivante permet de garantir que la conception réalisé répond aux besoins du système et que l’architecture en conséquence est maintenable et extensible.

Ce document met en collaboration tous les membres de l’équipe : architectes, concepteurs…

# Objectif du logiciel

Afin de contribuer à l’éradication de l’épidémie Covid’19, l’objectif est de réaliser une application de traçage d’accompagnement des citoyens.

Le projet a été lancée par l’Etat marocain début Avril 2020. Une telle application a déjà été développée par plusieurs pays notamment la Corée du sud, la Chine, le Ghana. Des ressources ont été partagées en open source et peuvent être utilisées pour optimiser le projet. Le respect des données personnelles (Loi n° 09-08 relative à la protection des personnes physiques à l’égard du traitement des données à caractère personnel) doit être pris en compte.

## Contexte

## La Covid-19 est une maladie infectieuse qui a touché plus de 4 millions personnes, causant la mort de plus 270 millions et ne cesse d’augmenter dans certaines pays malgré l’enregistrement d’un nombre record de guérisons

## Besoins fonctionnels

## prévenir immédiatement si vous avez été en contact rapproché avec une personne que vous ne connaissez pas et qui vient d’être testée positive au COVID-19

## Se diagnostiquer à travers un questionnaire .

## Besoins non fonctionnels

## L’application doit rappeler les consignes à respecter et les méthodes préconisées (lavage des mains, porte de masque, …)

## L’application doit permettre de connaitre l’état de l’épidémie dans le pays

# Structure

## Vue des couches

le propriétaire d’une vue de couche d’entités peut définir les champs ou entités disponibles dans la vue. On peut également limiter la vue de la couche d’entités hébergée à une zone donnée en définissant une étendue spatiale. Ces définitions sont enregistrées avec la vue de couche d’entités hébergées.

## Sous-systèmes et paquetages

## Interfaces

Les interfaces du coronAlert :

* Interface qui alertera l’utilisateur s’il était en contact avec une personne qui vient d’être testé positive un covid-19
* Interface montrant l’état de l’épidémie dans le pays
* Interface qui propose des questions
* Interface qui rappelle les consignes

# Comportement

## Réalisation des cas d’utilisation

Cas d’utilisation :

* Se notifier en cas de contact avec une personne infectée
* Détecter les zones de forte probabilité d’infection
* Se diagnostiquer à travers un questionnaire

## Mécanismes

On peut utiliser :

* Patron de conception : est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels
* Patron d’architecture : une solution générale et réutilisable à un problème d'architecture récurrent. Les patrons d'architecture sont semblables aux patrons de conception mais ont une portée plus large. Ils servent de modèle de référence et de source d'inspiration lors de la conception de l'architecture d'un système ou d'un logiciel informatique, pour décomposer celui-ci en éléments plus simples

# Autres vues

## Vue processus (optionnel)

## Vue implémentation (optionnel)

## Vue déploiement (optionnel)

## Vue données (optionnel)

# Concepts du domaine

Les concepts sont utilisé pour atteindre les objectifs ou solutions souhaitées dans la conception d’un projet et sont définis comme étant une réflexion qui assemble, combine ou intègre plusieurs éléments.

# Qualités de l’architecture

L’architecture utilisé : MVC (**Modèle-vue-contrôleur** )



# Points ouverts