
MADINA-TIC

RAPPORT DE CONCEPTION

Par: BELDJELTI Abdelatif

Date de création: 06/07/2019

Date de dernière modification: 30/07/2019

SOMMAIRE

TABLE OF CONTENTS

Sommaire.....	2
Introduction.....	3
Outils de modélisation.....	3
Draw.io.....	3
UML.....	3
Architectures adoptés.....	3
L'architecture MVC.....	3
.....	4
Modèle.....	4
Vue.....	4
Contrôleur.....	4
L'architecture 3 tiers.....	4
Diagramme UML.....	5
Diagramme composant-déploiement.....	5
Modèle relationnel.....	6
Maquettage.....	7
Conclusion.....	8

INTRODUCTION

Dans le cadre de notre projet « MADINA-TIC », nous devons concevoir une plateforme participative pour le signalement et la prise en charge des problèmes des citoyens afin d'améliorer leur cadre de vie.

Durant cette étape, on définit l'architecture adoptée à notre système puis on réalise le diagramme composant/déploiement. On présente ensuite la base de données en concevant le modèle relationnel et on clôture notre conception par la réalisation des maquettes IHM.

OUTILS DE MODÉLISATION

DRAW.IO

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter.

L'interface est simple et facile d'utilisation, les objets sont classés à gauche de l'écran par thèmes, une moteur de recherche vous permet d'interroger la base de données de dessins. Une fois votre travail terminé, la sauvegarde est réalisée au format XML (Diagramly XML Document), l'exportation propose différents formats (png, jpg, gif, svg, html, intégration en pages web...). Il est possible d'importer un diagramme au format

UML

UML (Unified Modeling Language) est une méthode de modélisation graphique, elle est utilisée en conception orienté objet et en développement logiciel pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement du système.

ARCHITECTURES ADOPTÉS

L'ARCHITECTURE MVC

Une application conforme au motif MVC comporte trois types de modules : les modèles, les vues et les contrôleurs.

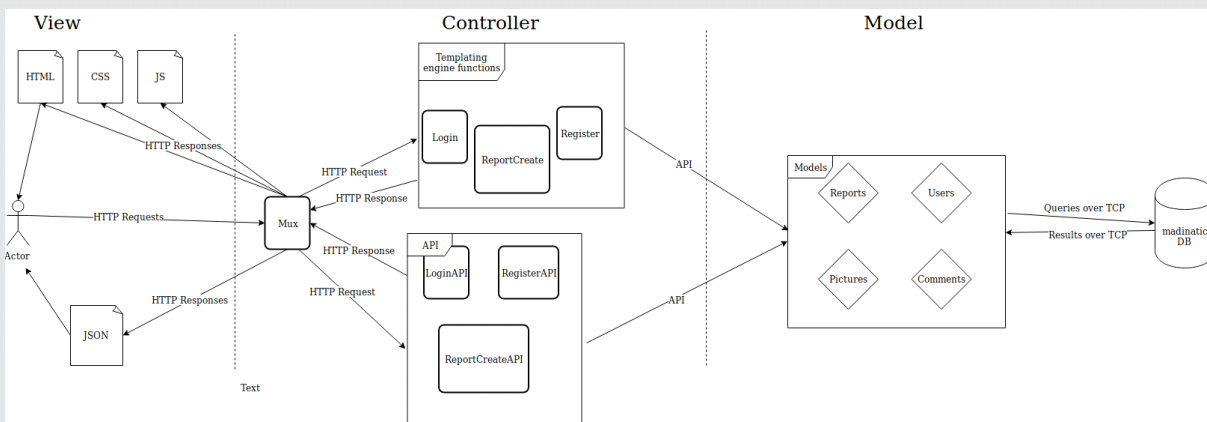


Figure 1. Architecture MVC

MODÈLE

Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Le modèle représente l'univers dans lequel s'inscrit l'application.

VUE

Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle.

CONTRÔLEUR

Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

L'ARCHITECTURE 3 TIERS

L'architecture logique de notre système est divisée en trois niveaux ou couches :

- Couche de présentation (Niveau 1) : l'affichage et les traitements locaux (contrôle de saisie, mise en forme e données...) sont pris en charge par le poste client.
- Couche de traitement (Niveau 2) : les traitements applicatifs globaux sont par le service applicatif (serveur d'application).
- Couche d'accès aux données (Niveau 3) : les services de base de données sont pris en charge par le serveur de données.

DIAGRAMME UML

DIAGRAMME COMPOSANT-DÉPLOIEMENT

Après avoir cité les différentes architectures relatives à notre système, nous passons maintenant à la modélisation de ces derniers à travers le diagramme composant-déploiement.

Le diagramme déploiement illustre la disposition physique des différents matériels (ou nœuds) qui entrent dans la composition du système et la répartition des composants (diagramme des composants) au sein des nœuds. Les composants sont des unités physiques d'empaquetage logiciel.

On présente ci-dessous le diagramme composant-déploiement de notre système:

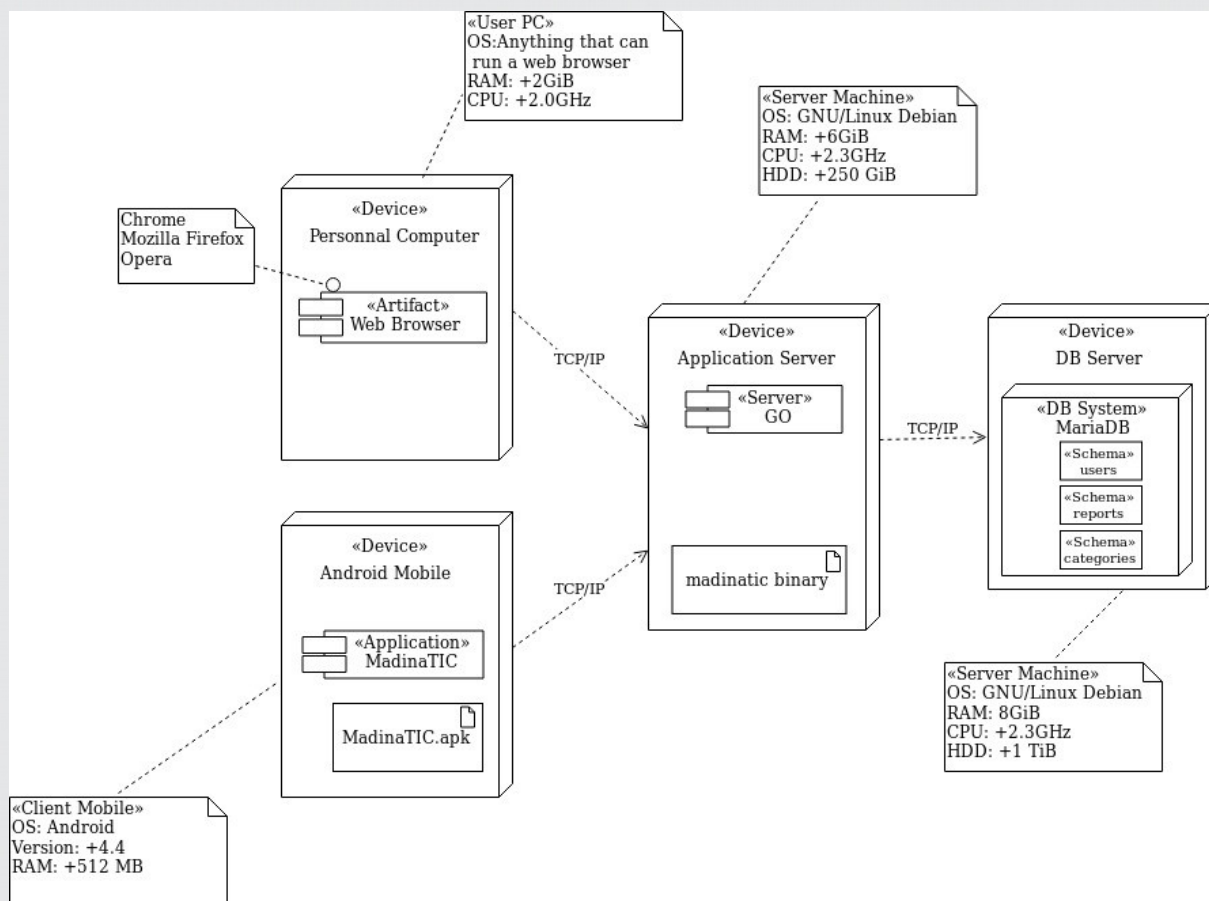


Figure 2. Diagramme composant-déploiement

MODÈLE RELATIONNEL

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles. Cette modélisation qui repose sur des principes mathématiques mis en avant par E.F. Codd est souvent retranscrite physiquement « implémentée » dans une base de données.

Le modèle relationnel de notre système est donné par le schéma des tables de la base de données suivantes :

- users(pk_userid, username, email, phone, password, confirm_token, reset_token, created_at, modified_at)
- citizens(pk_userid, first_name, family_name)
- authorities(pk_userid, name)
- categories(pk_catid, cat_name, #fk_userid)
- reports(pk_reportid, title, descr, created_at, modified_at, latitude, longitude, curr_state, state_modified_at, #fk_userid)
- pictures(pk_name, #fk_reportid)
- repcats(#fk_reportid, #fk_userid, solved)
- comments(#fk_reportid, #fk_userid, created_at, comment, modified_at)
- upvotes(#fk_reportid, #fk_userid)
- subs(#fk_reportid, #fk_userid)
- bans(#fk_userid, solved)
- notifications(#fk_reportid, #fk_userid, created_at, n_type, n_message)

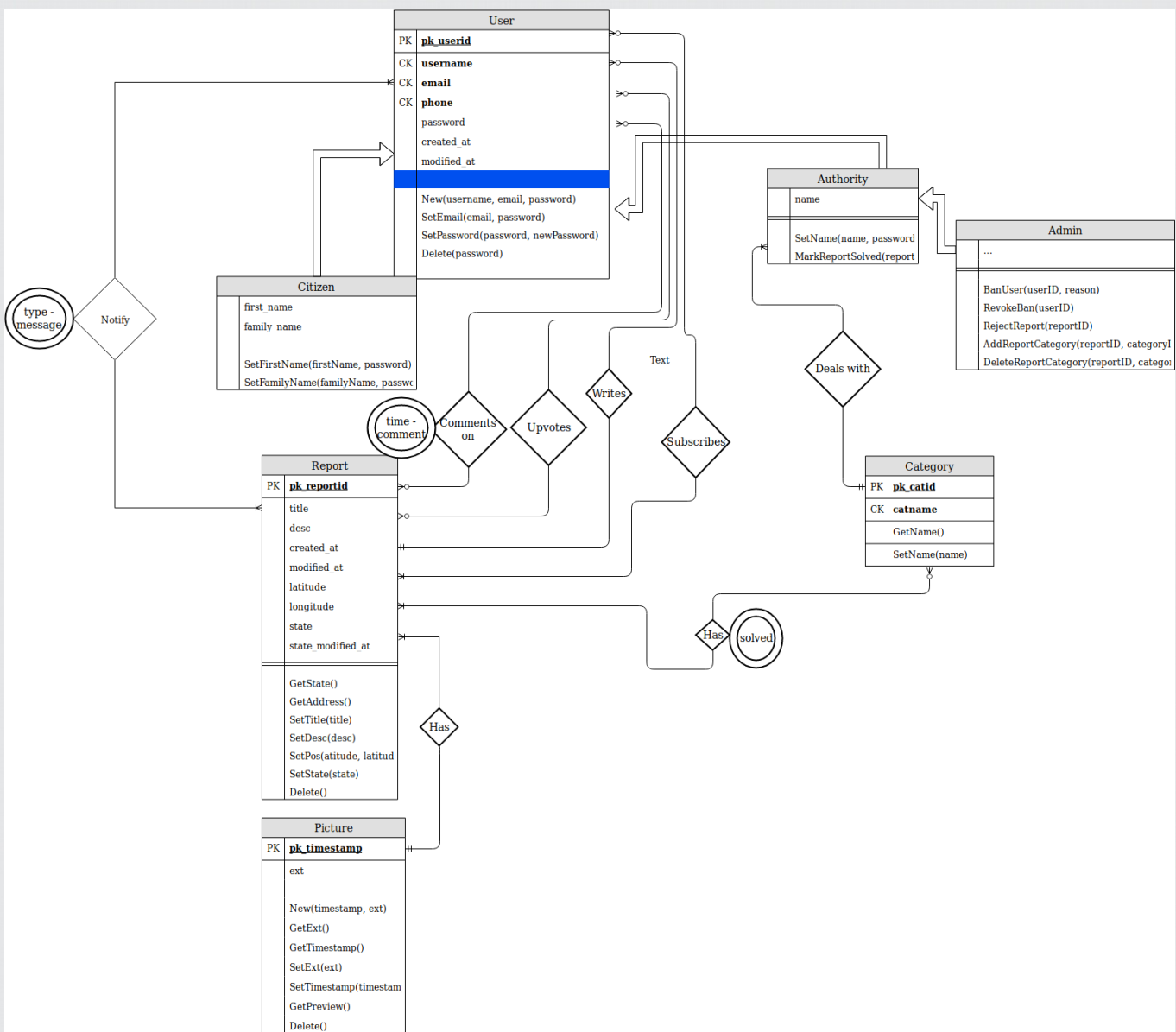


Figure 3. Le modèle relationnel E/A

MAQUETTAGE

En ergonomie, le maquettage/ prototypage consiste à simuler l'interface d'un produit, de façon complète ou partielle, afin d'obtenir des informations sur l'interaction des utilisateurs avec le futur produit. Effectué suffisamment tôt, le prototypage permet de détecter les problèmes d'utilisabilité et de vérifier les besoins des utilisateurs avant le codage complet. Cela favorise l'optimisation du processus et des coûts de développement.

Les interfaces de notre système sont présentées dans ce fichier PDF.

CONCLUSION

La conception de notre application nous a permis de définir la procédure de réaliser le projet d'une façon détaillée à travers la définition des architectures du système, et le passage au diagramme de composants/déploiement ensuite le modèle relationnel qui a mieux concrétisé et approfondi la compréhension de notre système. Enfin, le prototypage nous a permis d'identifier les interactions des utilisateurs avec le futur produit.