



UNIVERSITE INTERNATIONALE DE RABAT – ÉCOLE D'INFORMATIQUE ET
NUMERIQUE

**Rapport de stage d'été :
Conception et développement d'une application
Mobile de gestion des plaintes**

Réalisé par:

Hammani
Mohamed

Encadré par :

M. Youssef Zairi (HACA)
M. Najib Mehdi (UIR)

2021/2022

Remerciements

Comme le précise Michel Béaud dans son ouvrage «l'art de la thèse », aucun travail ne s'accomplit dans la solitude.

Je tiens donc à témoigner ma profonde gratitude à toutes les personnes qui ont participé de près ou de loin à l'élaboration de ce travail.

Je remercie M. BOUZMARNI Najib, directeur de Département Informatique à la Haut Autorité de la Communication Audio-visuel (HACA), de m'avoir permis d'effectuer mon stage d'été au sein de la Haute Autorité de la Communication Audiovisuelle.

Mes vifs remerciements vont à Mme. MOUJTAHID Amina, Chef de d'unité de développement Software, mon encadrant M. ZAIRI Youssef à la HACA et M. HAMMOUDA Nassim qui m'ont proposé un sujet très intéressant. Je leur suis très reconnaissant pour leurs directives précieuses et conseils pertinents qui m'ont été d'un appui considérable tout au long de mon stage.

Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de mes remerciements les plus chaleureux.

Résumé

Le présent document synthétise le travail effectué au sein de l'entreprise HACA au titre du projet de stage d'été, qui s'intitule « Conception et développement d'une application mobile de gestion des plaintes ».

L'analyse des besoins a permis l'identification des acteurs principaux et le développement des diagrammes de cas d'utilisation.

La conception a été traitée en mettant en place le diagramme de classe, diagramme de cas d'utilisation et l'architecture de l'application.

En ce qui concerne le développement de l'application, le framework Spring boot et Ionic ont été utilisés et moyennant les services web RestFul, la communication des données a été assurée.

Mots clés : Plaintes ,mobile, Spring Boot, Ionic, HACA, RestFul.

Liste des abréviations

HACA : Haute Autorité de la Communication Audiovisuelle

UML: Unified Modeling Language

DAO: Data access object

JPA: Java Persistence API

JDBC: Java Database Connectivity

SQL: Structured Query Language

HTTP: Hypertext Transfer Protocol

JSON: JavaScript Object Notation

HTML: Hypertext Markup Language

CSS : Cascading Style Sheets

JS: JavaScript

IoC: Inversion of control

SGBD : Système de gestion de base de données

MVC: Model-view-controller

Java EE: Java Enterprise Edition

DOM: Document Object Model

OS : Operating system

API : Application programming interface

Table des figures

1.1	la Haut Autorité de la Communication Audiovisuel	2
1.2	Organigramme de la HACA	3
2.1	Diagramme de cas d'utilisation.....	6
2.2	Diagramme de classe.....	8
2.3	Modèle de l'architecture de l'application	9
3.1	Java	10
3.2	Intellij.....	11
3.3	Maven	11
3.4	Spring Boot.....	11
3.5	Gitlab.....	12
3.6	Bootstrap.....	12
3.7	Ionic	12
3.8	Microsoft SQL Server.....	13
3.9	Hibernate	13
3.10	Spring Data	14
3.11	Page d'accueil d'un demandeur	14
3.12	Interface de déposition d'une Plainte	14
3.13	Interface de visualisation des plaintes	15

Liste des tableaux

2.1 Tableau des acteurs en interaction avec le système5

Table des matières

Introduction générale	1
1 Cadre général du projet	2
Introduction.....	2
1.1 Présentation de l'organisme d'accueil	2
1.1.1 La Haute Autorité de la Communication Audiovisuelle	2
1.1.2 L'unité informatique	3
1.1.2.1 L'unité de Développement Softwares	3
1.1.2.2 L'unité d'exploitation des systèmes techniques	3
1.1.2.3 L'unité production des données	4
1.2 Contexte du projet	4
1.2.1 Problématique	4
1.2.2 Objectif	4
Conclusion	4
2 Analyse et conception	5
Introduction.....	5
2.1 Analyse des besoins.....	5
2.1.1 Identification des acteurs	5
2.1.2 Diagrammes des cas d'utilisation	5
2.2 La conception de l'application.....	7
2.2.1 Diagramme de classe	7
2.2.2 Architecture de l'application.....	8
Conclusion	9
3 Réalisation	10
Introduction.....	10
3.1 Outils de développement.....	10
3.1.1 Environnement de développement.....	10
3.1.2 Framework.....	11
3.1.3 Bibliothèque.....	12
3.1.4 Outils de base de données	13
3.2 Interfaces de l'application	14
Conclusion	17
Conclusion générale	18

Introduction générale

Dans le secteur audiovisuel, les programmes de télévision et de radio peuvent recourir à des moyens désespérés pour attirer l'attention. Pour éviter de tels actes, on leur donne une ligne directrice à suivre, mais que se passe-t-il si un téléspectateur/auditeur remarque quelque chose qui le dérange ou l'offense, comment peut-il exprimer son désaccord ou sa désapprobation, nous devons avoir donc une application qui lui permette de se plaindre.

Dans ce contexte se situe mon projet de stage d'été de première année que j'ai effectué au sein de la Haut Autorité de la Communication Audio-visuel (HACA), mon projet est intitulé : La conception et la réalisation d'une application mobile de gestion des plaintes au sein de la HACA.

L'objectif de mon projet sera donc de concevoir et réaliser cette application mobile, tout utilisant les nouvelles technologies que j'ai eu l'opportunité de découvrir et qui sont Spring boot, Ionic et Microsoft SQL Server.

Afin d'atteindre mon objectif, j'ai suivi le plan suivant : premièrement, j'ai introduit le cadre général du projet en présentant l'organisme d'accueil ainsi que l'unité informatique puis le contexte du projet, deuxièmement, j'ai procédé à l'analyse et la conception de l'application à réaliser et finalement j'ai présenté les outils de développement retenus Ainsi que les interfaces et fonctionnalités de l'application réalisée.

Chapitre 1

Cadre général du projet

Introduction

Ce chapitre présente tout d'abord d'une manière générale l'organisme d'accueil pour enchaîner avec une contextualisation du projet. Cette partie du rapport a pour objectif d'une part de donner un aperçu global sur l'environnement dont lequel j'ai effectué mon stage, d'autre part de décrire le contexte du projet, sa problématique et son objectif.

1.1 Présentation de l'organisme d'accueil

1.1.1 La Haute Autorité de la Communication Audiovisuelle

La réforme du paysage audiovisuel national est passée du statut d'ambition à celui de réalisations concrètes. Elle traduit la ferme volonté du Maroc de développer l'option démocratique à travers la consécration du pluralisme, la consolidation des fondements de l'état de droit et des institutions et la garantie de l'exercice de la liberté d'expression et d'opinion, dans un esprit de responsabilité.

La création de la Haute Autorité de la Communication Audiovisuelle, la suppression du monopole de l'état en matière de radio et de télévision et la promulgation de la loi relative à la communication audiovisuelle constituent en effet des jalons importants dans le processus visant à moderniser le secteur public audiovisuel, à favoriser la création d'entreprises privées de radio et de télévision dans un cadre normatif et institutionnel clair et réfléchi, et de consacrer ainsi, dans les faits, la double liberté de communication et d'entreprise, à travers une libéralisation régulée par une autorité administrative indépendante.[1]



الهيئة العليا للاتصال السمعي البصري
ⵎⴰⵔⴰⵏ ⵉⵎⴰⵔⴰⵏ ⵉⵎⴰⵔⴰⵏ ⵉⵎⴰⵔⴰⵏ
Haute Autorité de la Communication Audiovisuelle

FIGURE 1.1 – Logo de la HACA

La Haute Autorité de la Communication Audiovisuelle, créé par le Dahir n°1-02-212 du 31 août 2002 est une autorité administrative indépendante. Elle est composée de deux organes :[2]

1. Le Conseil Supérieur de la Communication Audiovisuelle (CSCA), composé de neuf membres dont la Présidente. Ce conseil a pour attributions notamment de :
 - Veiller au respect, par tous les pouvoirs ou organes concernés, des lois et règlements applicables à la communication audiovisuelle ;
 - Instruire les demandes d'autorisation de création et d'exploitation des entreprises de communication audiovisuelle ;

- Accorder les autorisations d'utilisation des fréquences nécessaires au fonctionnement des réseaux de diffusion des radios et télévisions publiques et privées ;
 - Contrôler le respect, par les organismes de communication audiovisuelle, du contenu des cahiers de charges et, de manière générale, le respect, par lesdits organismes des principes et règles applicables au secteur ;
 - Veiller au respect de l'expression pluraliste des courants de pensée et d'opinion, notamment en matière d'information politique, tant par le secteur privé que par le secteur public de l'audiovisuel ;
2. La Direction Générale de la Communication Audiovisuelle (DGCA), qui, sous l'autorité de la Présidente, est chargée notamment de l'exécution des délibérations de CSCA, de l'administration et de la gestion des services et du personnel administratif et technique de la HACA. Elle est dirigée par un Directeur Général et se compose des structures en charge des attributions suivantes :
- Suivi des programmes
 - Infrastructures techniques et veille technologique
 - Etude et développement
 - Etudes juridiques
 - Système d'information
 - Coopération Internationale
 - Affaires administratives et financières
 - Secrétariat du conseil.
 - Audit et contrôle de gestion
 - Documentation

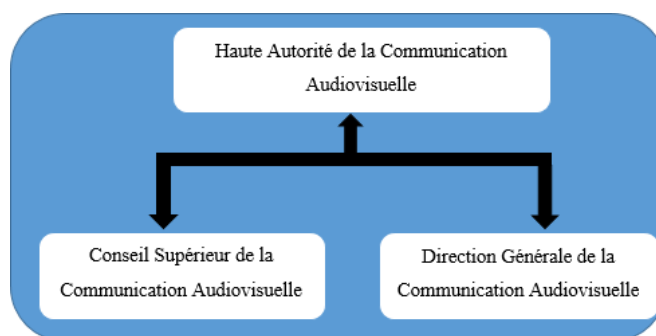


FIGURE 1.2 – Organigramme de la HACA

1.1.2 Département Systèmes d'Information

Le département Systèmes d'Information s'articule autour de trois unités [2] :

1.1.2.1 L'unité de Développement Softwares

Cette unité a pour rôle de :

- Concevoir des projets informatiques et en assurer le développement selon les règles de l'art et les choix techniques et méthodologiques adoptés par la HACA ;
- Implémenter ces projets informatiques, en conformité avec les orientations stratégiques et opérationnelles de la HACA, assurer leur maintenance et leur adaptation aux évolutions de l'Institution et de l'environnement technique ;
- Concevoir des projets informatiques et en assurer le développement selon les règles de l'art et les choix techniques et méthodologiques adoptés par la HACA ;
- Implémenter ces projets informatiques, en conformité avec les orientations stratégiques et opérationnelles de la HACA, assurer leur maintenance et leur adaptation aux évolutions de l'institution et de l'environnement technique ;
- Mettre en place les progiciels de gestion en coordination avec les départements et unités concernées ;
- Produire les manuels techniques de maintenance et élaborer les guides et tutoriels des applications développées en interne.

1.1.2.2 L'unité d'exploitation des systèmes techniques

Parmi les rôles de l'unité d'exploitation des systèmes techniques au sein du département Systèmes d'Information :

- Assurer la bonne marche, l'exploitation et la maintenance de l'infrastructure technique (informatique, télécommunications, acquisition et stockage des émissions télévisuelles et radiophoniques) tant au niveau du siège de la HACA que des régions ;
- Administrer et superviser les réseaux techniques et les bases de données y afférentes ;
- Mettre en place les ressources bureautiques et en assurer le suivi et la maintenance.

1.1.2.3 L'unité production des données

L'unité production des données s'occupe d'un ensemble de services :

- Recenser, auprès des différentes entités de la HACA et spécifiquement du Département des Etudes et de Suivi des Programmes, les besoins en matière de production de données ;
- Produire les résultats nécessaires à l'élaboration des relevés pluralisme relatifs au respect du temps de parole et d'antenne, selon la base des règles édictées par la HACA en la matière ;
- Produire les données relatives aux obligations quantitatives concernant la programmation des supports audiovisuels, et faire ressortir les statistiques relatives à la cartographie de la programmation des supports audiovisuels ;
- Produire aux différentes entités de la HACA les données dont ils auront besoin dans le cadre de leurs activités.

I.2 Contexte du projet

1.2.1 Problématique

Dans le secteur audiovisuel, les programmes de télévision et de radio peuvent recourir à des moyens désespérés pour attirer l'attention. Pour éviter de tels actes, on leur donne une ligne directrice à suivre, mais que se passe-t-il si un téléspectateur/auditeur remarque quelque chose qui le dérange ou l'offense, comment peut-il exprimer son désaccord ou sa désapprobation, nous devons avoir donc une application qui lui permette de se plaindre.

1.2.2 Objectif

L'objectif de ce projet sera donc la conception et le développement d'une application mobile dynamique de gestion des plaintes au sein de la HACA.

Conclusion

Ce chapitre a permis de situer le contexte général du projet, de spécifier la problématique, l'objectif du projet ainsi que le cahier des charges. A ce niveau, il y a lieu de s'interroger sur les besoins auxquels ce projet doit répondre, c'est ce qui va être détaillé dans le chapitre suivant.

Chapitre 2

Analyse et conception

Introduction

Ce chapitre présente, en premier lieu, les étapes suivies pour l'analyse des besoins du projet et la modélisation fonctionnelle avec les diagrammes UML. En second lieu, il définit la conception du projet. Après l'identification des différentes fonctionnalités de l'application dans le troisième chapitre. La deuxième partie du présent chapitre aborde l'étude conceptuelle de l'application proposée.

2.1 Analyse des besoins

L'analyse des besoins est une étape très importante dans le processus de l'étude et le développement des systèmes d'informations. Cette partie identifie l'ensemble des acteurs qui interagissent avec le système et définit l'ensemble des cas d'utilisation de ce dernier en se basant sur les diagrammes UML.

2.1.1 Identification des acteurs

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système. L'analyse du présent projet commence par une identification des acteurs agissants sur les différentes parties du système. Le tableau en dessous récapitule les acteurs en interaction avec le système en spécifiant les fonctions de chacun de ces derniers avant de définir plus précisément leurs interactions avec le système en utilisant des diagrammes UML.

Acteur	Fonction
Télespectateur /auditeur	Il peut déposer des Plaintes selon leur type.

TABLE 2.1 – Tableau des acteurs en interaction avec le système

2.1.2 Diagrammes des cas d'utilisation

Cette partie présente l'ensemble des cas d'utilisation de l'application en se basant sur les diagrammes des cas d'utilisation du système.

Diagramme des cas d'utilisation du demandeur :

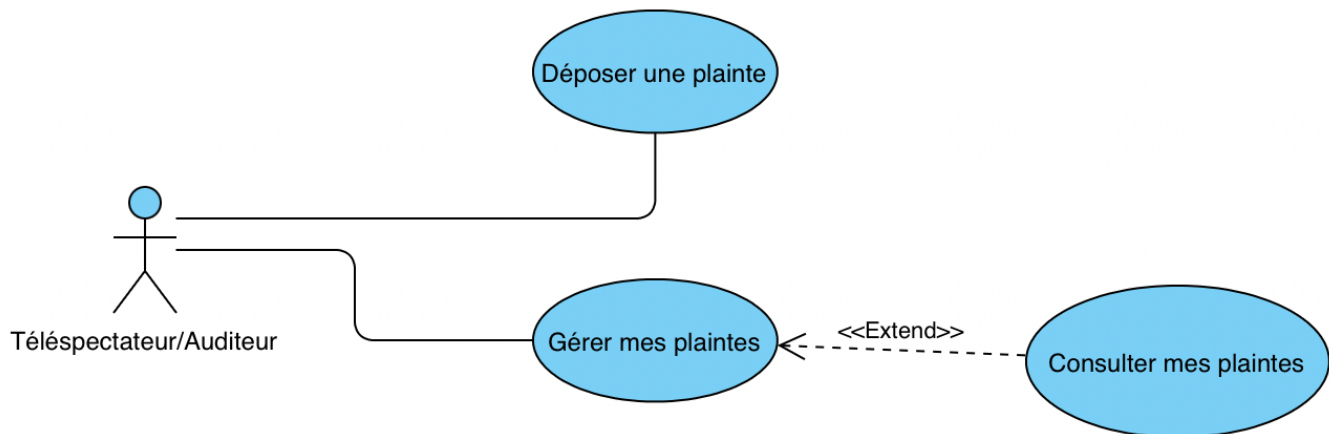


FIGURE 2.1 - Diagramme de cas d'utilisation du demandeur

2.2 La conception de l'application

2.2.1 Diagramme de classe

Dans ce qui suit, une présentation du diagramme de classe de l'application. Ce dernier représente le noyau du système et tous les traitements métiers liés au projet.

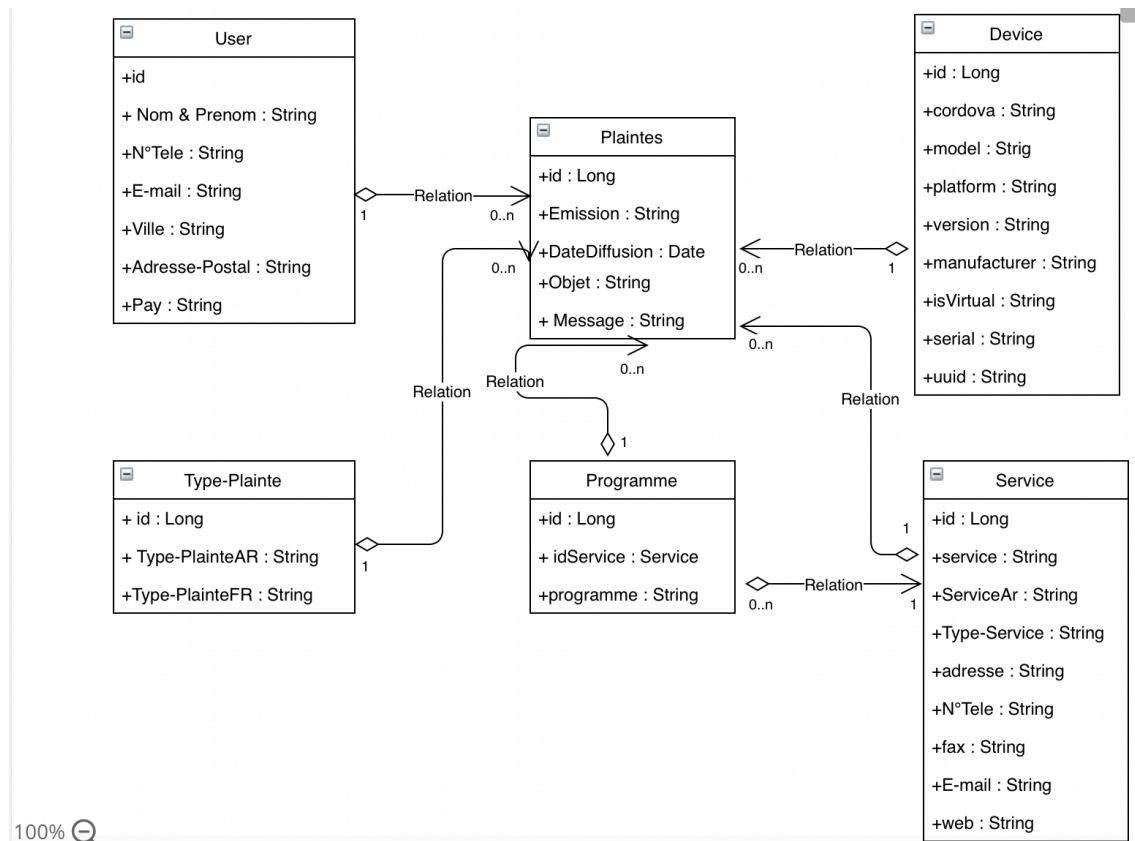


FIGURE 2.2 – Diagramme de classe

2.2.2 Architecture de l'application

L'architecture de notre application s'intéresse au découpage logique et la façon de regrouper les composants selon le type de la fonction et le traitement qu'ils effectuent. Pour notre application, nous avons opté pour une architecture multicouche.

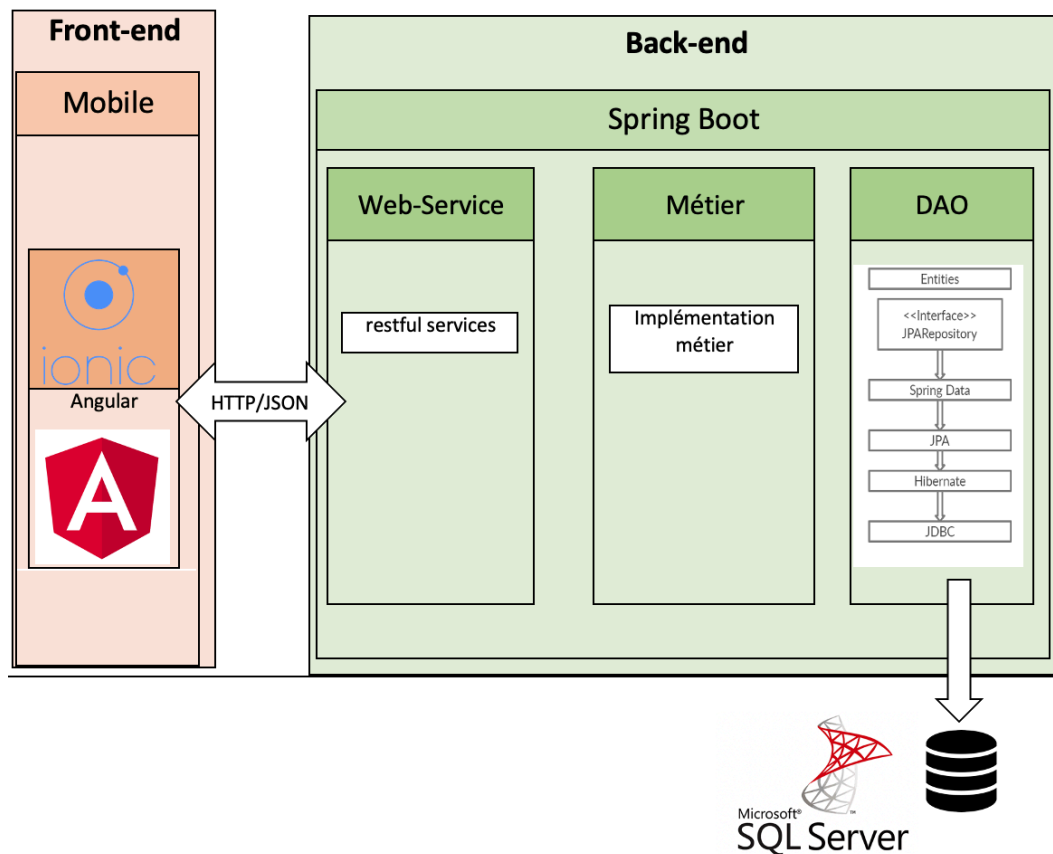


FIGURE 2.5 – Modèle de l'architecture de l'application

Couche présentation (Client) : correspond à la partie de l'application visible et interactive avec les utilisateurs. Elle communique avec la couche métier par les web services et le protocole HTTP.

Couche web-service : il s'agit de la couche responsable du lien entre la couche de présentation et la couche métier. Toutes les requêtes sont exécutées par la méthode POST pour renforcer la sécurité des web-services.

Couche métier (Business Layer) : correspond à la partie fonctionnelle de l'application responsable de l'implémentation de la « logique ». Celle-ci décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.

Couche accès aux données (Data Layer) : chargée de l'accès aux données et de leur manipulation, indépendamment du SGBD choisi.

Conclusion

Ce chapitre présente une vue conceptuelle de l'application à mettre en place. Ainsi, il expose les différents diagrammes UML pour mieux comprendre les fonctionnalités offertes et mieux représenter la communication entre les différentes entités du projet. Le chapitre suivant, présente la partie mise en œuvre de l'application.

Chapitre 3

Réalisation

Introduction

Nous allons présenter dans ce chapitre les détails de l'implémentation. Tout d'abord, une présentation des technologies et outils utilisés, ensuite une présentation de quelques interfaces de l'application réalisée.

3.1 Outils de développement

Cette partie énumère l'ensemble des technologies utilisées dans ce projet.

3.1.1 Environnement de développement

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.[3]



FIGURE 3.1 - Java

IntelliJ IDEA est un IDE, autrement dit un ensemble d'outils destinés au développement logiciel. Contrairement à Eclipse qui est géré par un consortium (Fondation Eclipse), IntelliJ IDEA est édité par une société commerciale (JetBrains) qui en assure le développement et le support depuis janvier 2001. IDEA existe en deux versions : Community et Ultimate.

Pour faire simple, la version Community, gratuite, est avant tout destinée au développement d'applications « lourdes » Java, Scala et Android. Dès qu'il s'agit de développer des applications web, il faut se tourner vers l'édition Ultimate. La version Community gère nativement les langages suivants : Java, Scala, Groovy, Clojure et XML, XSD et DTD. Avec la version Ultimate s'ajoutent les langages dédiés au développement web, à savoir le HTML, CSS, JavaScript, CoffeeScript, ActionScript. Viennent également le support du Freemarker, de Velocity, du XSL, XPath, SQL, Ruby et JRuby, Python ou encore PHP. Certains de ces langages nécessitent toutefois l'ajout de plugins gratuits. Avec la version Ultimate vient également le support des frameworks les plus courants pour le développement autour de l'écosystème de la JVM ou du web (Spring, Playframework, JavaEE 6, GWT, Hibernate).[4]

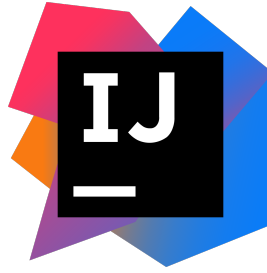


FIGURE 3.2 -IntelliJ

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine.

Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. Il permet notamment d'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet et aussi de gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet et de générer des documentations concernant le projet.[5]



FIGURE 3.3 -Maven

3.1.2 Frameworks

Spring Boot est un nouveau framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications Spring.

Le framework propose une approche dogmatique de la configuration, qui permet d'éviter aux développeurs de redéfinir la même configuration à plusieurs endroits du code. Dans ce sens, Boot se veut d'être un acteur majeur dans le secteur croissant du développement d'applications rapide.

Pourtant, Spring Boot n'est pas une alternative à ces nombreux projets qui incluent la couche "Fondation" de la plate-forme Spring IO. L'objectif de Spring Boot n'est pas d'apporter de nouvelles solutions pour les nombreux problèmes qui ont déjà été résolus, mais plus d'influencer la plate-forme en favorisant une expérience de développement qui simplifie l'utilisation de technologies déjà existantes. Ceci fait de Boot un choix idéal pour les développeurs qui sont familiers avec l'écosystème Spring, mais aussi pour les nouveaux adoptants en leur permettant d'appréhender les technologies de Spring de manière simplifiée.[6]



FIGURE 3.4 -Spring Boot

GitLab est une plateforme de développement collaborative open source éditée par la société américaine du même nom. Elle couvre l'ensemble des étapes du DevOps. Se basant sur les fonctionnalités du logiciel Git, elle permet de piloter des dépôts de code source et de gérer leurs différentes versions. Son usage est particulièrement indiqué pour les développeurs qui souhaitent disposer d'un outil réactif et accessible.



FIGURE 3.5 –GitLab

Bootstrap est également un framework de développement Web populaire utilisé pour créer des sites Web. Il a été développé par une équipe sur Twitter et est un projet open source depuis 2011. Le framework Bootstrap comprend des styles CSS, des bibliothèques Javascript et des fichiers HTML. Bootstrap permet de créer facilement des sites Web réactifs plutôt que de les concevoir de toutes pièces. Il distingue par le développement de sites Web mobiles et réactifs. Avec des connaissances de base en HTML et CSS, on peut créer des formulaires, tableaux, boutons, typographie, navigation, modaux, carrousels d'images et plugins Javascript facultatif, en utilisant les modèles prédéfinis que Bootstrap fournit.[8]



FIGURE 3.6 –Bootstrap

3.1.3 Framework

Ionic est un framework libre d'utilisation qui permet de créer des applications mobiles pour iOS, Android et Windows Phone, à partir d'une base de code unique. En d'autres termes, Ionic est un outil de développement mobile multiplateforme.

Ionic a été fondé en 2012 par une start-up technologique appelée Drifty Co, et c'est aujourd'hui le meilleur framework open-source au monde pour la création d'applications mobiles (hybrides) multiplateformes.



FIGURE 3.7 –Ionic

3.1.4 Outils de base de données

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.[10]



FIGURE 3.8 -Microsoft SQL Server

Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java. Elle repose essentiellement sur l'utilisation des annotations, introduites dans Java 5. Elles permettent de définir très facilement, et précisément des objets métier, qui pourront servir d'interface entre la base de données et l'application.[11]

Hibernate est un Framework de mapping Objet-Relationnel, il assure la sauvegarde et la mise à jour d'objets Java dans des bases de données relationnelles. Les caractéristiques d'HIBERNATE sont les suivantes : Il fonctionne sur de nombreuses bases (MySQL, DB2, Oracle, PostgreSQL, etc.). Il repose sur une description déclarative de l'appariement entre les objets et les classes. Il implique un surcoût négligeable au niveau des traitements. Comme il gère les contextes transactionnels. Il offre un langage de requête objet proche syntaxiquement de SQL. Il est déployé dans de nombreux environnements industriels.[12]



FIGURE 3.9 -Hibernate

Spring Data a pour mission de fournir un modèle de programmation familier et cohérent basé sur Spring pour l'accès aux données tout en conservant les traits spéciaux du magasin de données sous-jacent. Il facilite l'utilisation des technologies d'accès aux données, des bases de données relationnelles et non relationnelles, des cadres de réduction de la carte et des services de données basés sur le cloud. Il s'agit d'un projet de parapluie qui contient de nombreux sous-projets spécifiques à une base de données donnée. Les projets sont développés en travaillant en collaboration avec de nombreuses entreprises et développeurs derrière ces technologies passionnantes.[13]



FIGURE 3.10 -Spring Data

3.2 Interfaces de l'application

Cette partie introduit l'application mobile réalisée pendant mon stage. je vais la présenter en des captures d'écrans.

Cette première interface représente la page d'accueil.

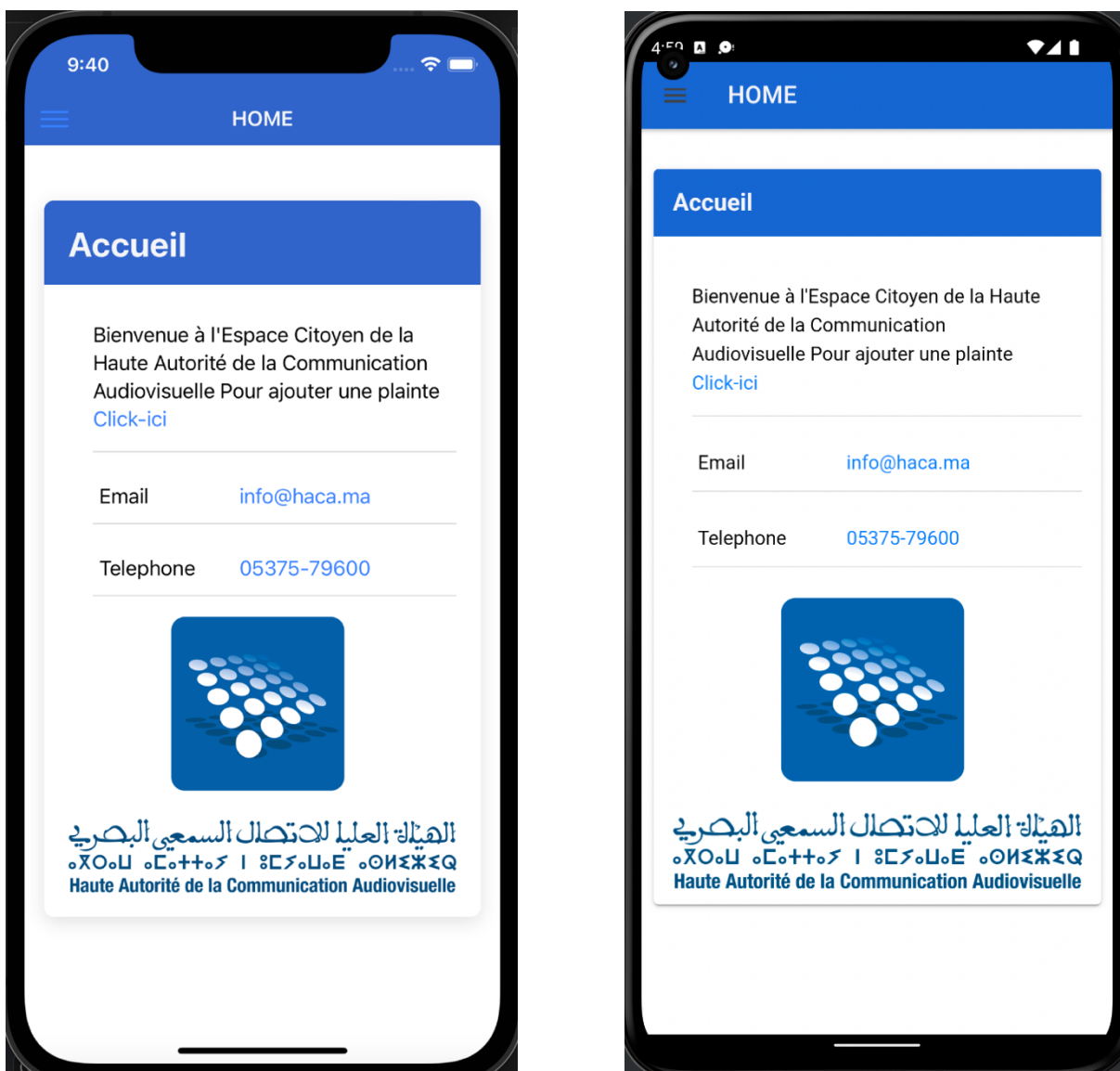
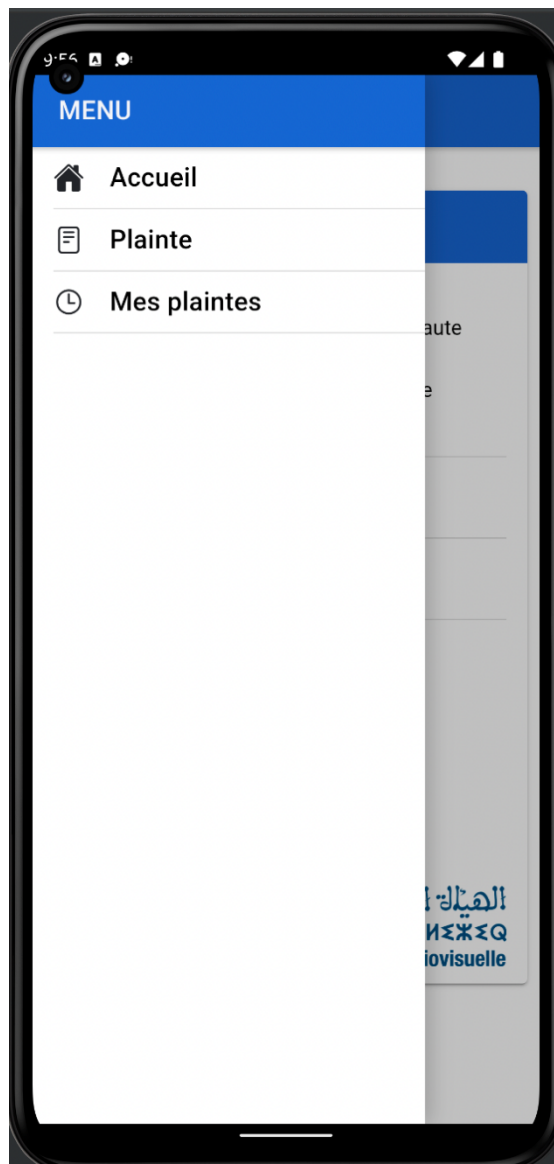
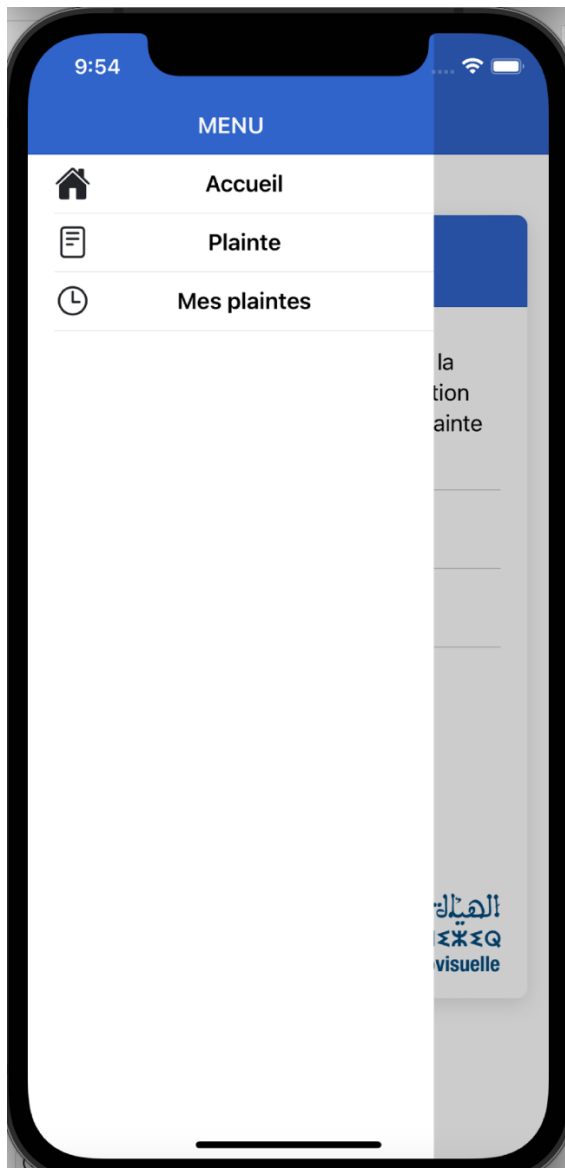


FIGURE 3.11 - Page d'accueil

Un utilisateur possède un menu avec trois liens, « Page d'accueil », « Mes plaintes » où il peut visualiser toutes ses plaintes et « Plainte » pour formuler une réclamation.



9:40

PLAINTÉ

Deposer une plainte

Type de Plainte type

Service service

Emission emission

Nom nom

Prenom prenom

N° Téléphone 0600000000

E-mail mail

Adresse postale adresse

Ville ville

Pays pay

Objet objet

Message message

Date 12/08/2022, 10:00

4:44

PLAINTÉ

Deposer une plainte

Type de Plainte type

Service service

Emission emission

Nom nom

Prenom prenom

N° Téléphone 0600000000

E-mail mail

Adresse postale adresse

Ville ville

Pays pay

Objet objet

Message message

Date 11/08/2022, 10:00

9:43

PLAINTÉ

Deposer une plainte

Type de Plainte type

Service service

Type de Plainte

Type 1

Type 2

Type 3

Cancel OK

Adresse postale adresse

Ville ville

Pays pay

Objet objet

Message message

Date 12/08/2022, 10:00

9:44

PLAINTÉ

Deposer une plainte

Type de Plainte type

Service service

Service

HitRadio

URadio

Radio 1

Radio 2

Radio 3

Cancel OK

Objet objet

Message message

Date 12/08/2022, 10:00

Ajouter

FIGURE 3.12 – Interface de déposition d'une Plainte

Chaque demande possède un type de plainte, service, émission, nom, prénom, N° Téléphone, mail, adresse postale, ville, pays, objet, Message et une date, dès que l'utilisateur a rempli la plainte, il sera redirigé vers la page « Mes plaintes ».

Pour visualise les plaintes soumit par l'utilisateur la page suivante s'affiche.

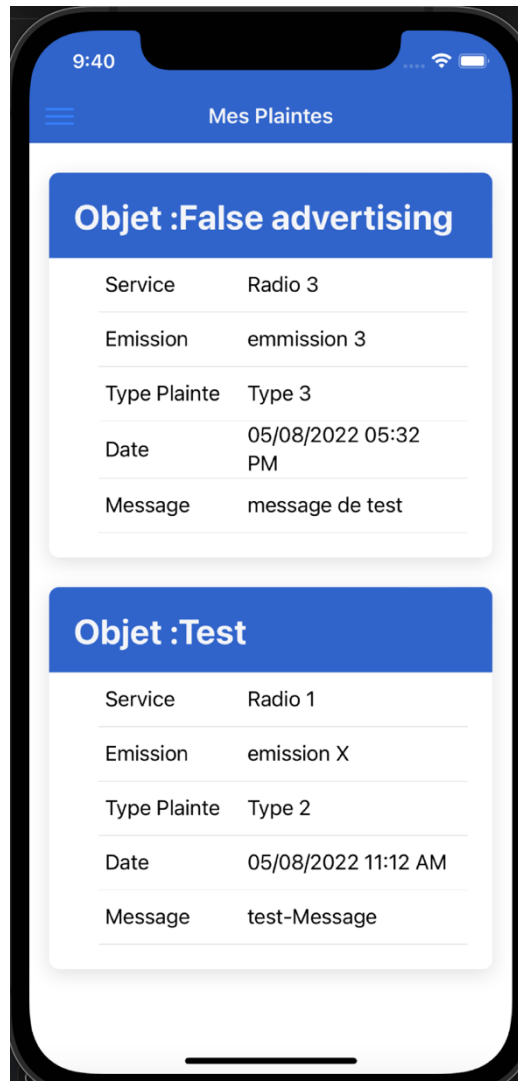


FIGURE 3.13 – Interface de visualisation des plaintes

Conclusion

Dans ce chapitre, on a vu les technologies utilisées dans notre projet ainsi que quelques pratiques pour la réalisation. Enfin, on a présenté le résultat sous forme de captures d'écran des interfaces homme-machine de l'application.

Conclusion générale

L'objectif de ce projet était la réalisation d'une application Web, qui permet la gestion des plaintes au sein de la HACA. Pour l'implémentation de l'application le choix technique s'est tourné vers Spring Boot pour la partie backend, vers Ionic pour la partie frontend et vers Microsoft SQL Server comme gestionnaire de base de données.

En termes de perspective, l'application a encore besoin d'améliorations pour assurer d'avantage une bonne gestion et pour faciliter et automatiser la tâche pour plus d'efficacité.

Pour conclure, on peut dire que cette expérience a été unique, car c'est une première expérience de prise en main d'une application mobile avec lesdites technologies.

Bibliographie

- [1] HACA, <https://www.refram.org/Les-membres/HACA-Maroc> .
- [2] HACA, <https://haca.ma/> .
- [3] Java, [https://fr.wikipedia.org/wiki/Java\(langage\)](https://fr.wikipedia.org/wiki/Java(langage)) .
- [4] IntelliJ, <https://www.ksup.org/communaute-/pourquoi-intellij-idea-101436.kjsp> .
- [5] Maven, <https://www.jmdoudoux.fr/java/dej/chap-maven.htm> .
- [6] Spring Boot, <https://www.infoq.com/articles/microframeworks1-spring-boot/> .
- [7] Spring, <https://spring.io/> .
- [8] Bootstrap, [https://fr.wikipedia.org/wiki/Bootstrap\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap(framework)) .
- [9] Ionic, [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))
- [10] Microsoft SQL Server, <https://fr.wikipedia.org/wiki/MicrosoftSQLserver> .
- [11] JPA, https://fr.wikipedia.org/wiki/JavaP_ersistenceAPI .
- [12] Hibernate, <https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm> .
- [13] Spring Data, <https://www.codeflow.site/fr/article/org-springframework>