



Dossier de conception

>> Diginamic.fr

Révisio n	Rédacteurs	Date	Objet
1	R. BONNAMY	17/02/2021	Création du document
2	F. Y. Henry	15/07/2023	Adaptation au projet

--	--	--	--

1 INTRODUCTION

1.1 *Objet du document*

Ce document a pour objectif de présenter l'essentiel des questions techniques liées à la mise en place de l'application Projet qualité de l'air.

Ce document présente :

- Le diagramme de classes
- Le modèle physique de données

2 SOMMAIRE

Table des matières

1 INTRODUCTION	3
1.1 Objet du document	3
2 SOMMAIRE	4
2.1.1 Table des matières.....	4
3 ARCHITECTURE LOGICIELLE	5
3.1 Produits et versions	5
3.1.1 Langages, frameworks et librairies spécifiques.....	5
3.1.2 Serveur de base de données.....	5
4 FOCUS TECHNIQUE	6
4.1 Diagramme de classes métier	6
4.2 Modèle physique de données	7
4.3 Règles de développement coté back	7
1 Règles de développement.....	7
5 TESTS ET INTÉGRATION	9
5.1 Stratégie de tests	9
5.2 Indicateurs de qualité de code	9

3 ARCHITECTURE LOGICIELLE

3.1 Produits et versions

Langages, frameworks et bibliothèques spécifiques

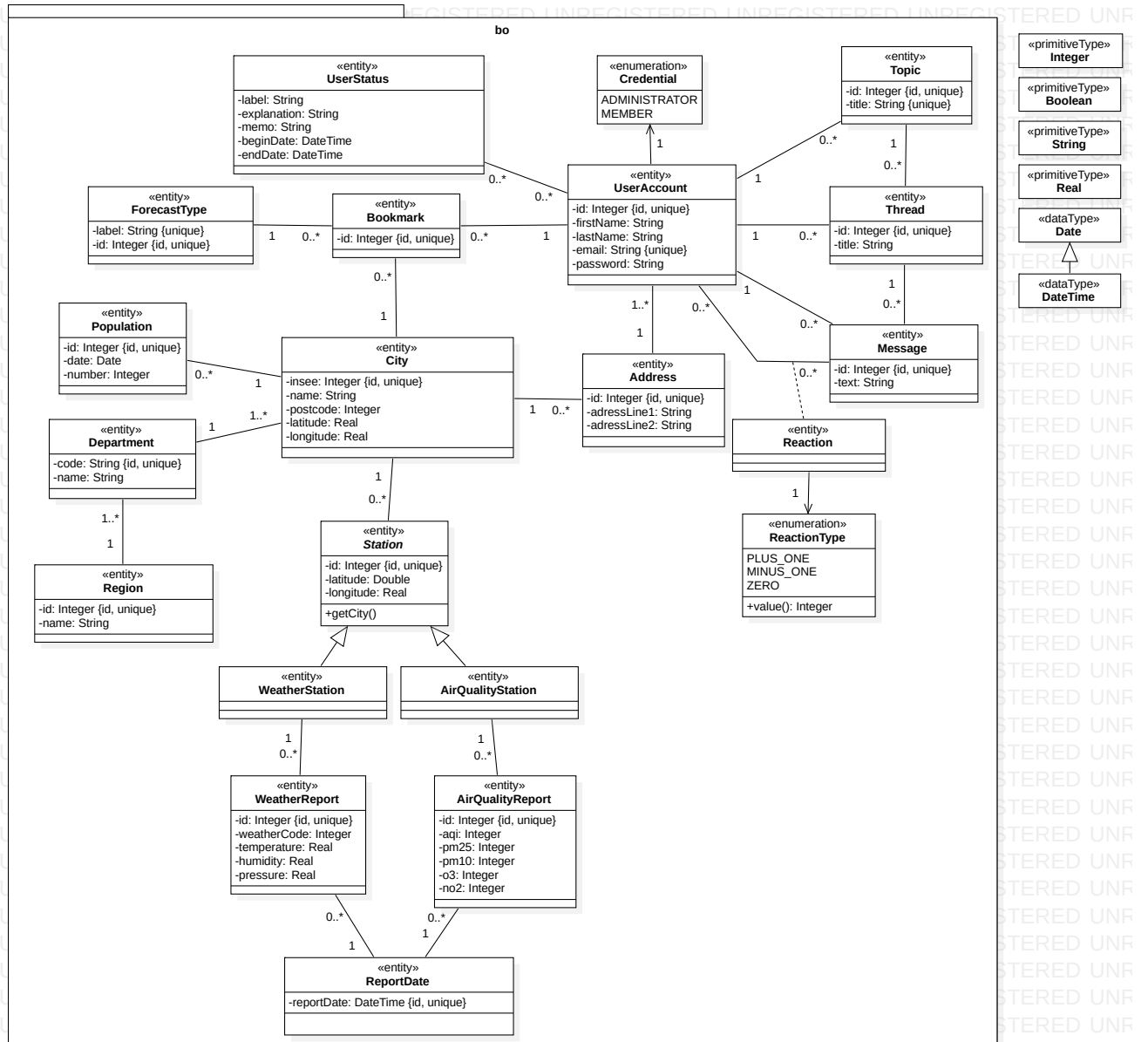
<u>Nom</u>	<u>Version</u>	<u>Référence</u>
JDK (OpenJDK)	17.0.7	https://www.oracle.com/java/
Spring Boot	3.1.1	https://spring.io/projects/spring-boot
JPA (Hibernate Core)	6.2.6.Final	https://hibernate.org/orm/
Angular	16.1.4	https://angular.io/
iText7 Core (génération de PDF)	8.0.0	https://kb.itextpdf.com/home/it7kb
LibreOffice	7.5.4	https://www.libreoffice.org/
StarUML	5.1.0	https://staruml.io/
Looping	4.0	https://www.looping-mcd.fr/
Figma	21 juin 2023	https://www.figma.com/
Leaflet	1.9.4	https://leafletjs.com/
Météo Concept	1.14	https://api.meteo-concept.com/
WAQI	1.5.0	https://aqicn.org/
API Découpage administratif	13 juillet 2023	https://geo.api.gouv.fr/decoupage-administratif
Git	2.30.2	https://git-scm.com/
Github (Forge logicielle)		https://github.com/
Apache Maven	3.8.1	https://maven.apache.org/
JUnit 5 (Jupiter)	5.9.3	https://junit.org/junit5/

Serveur de base de données

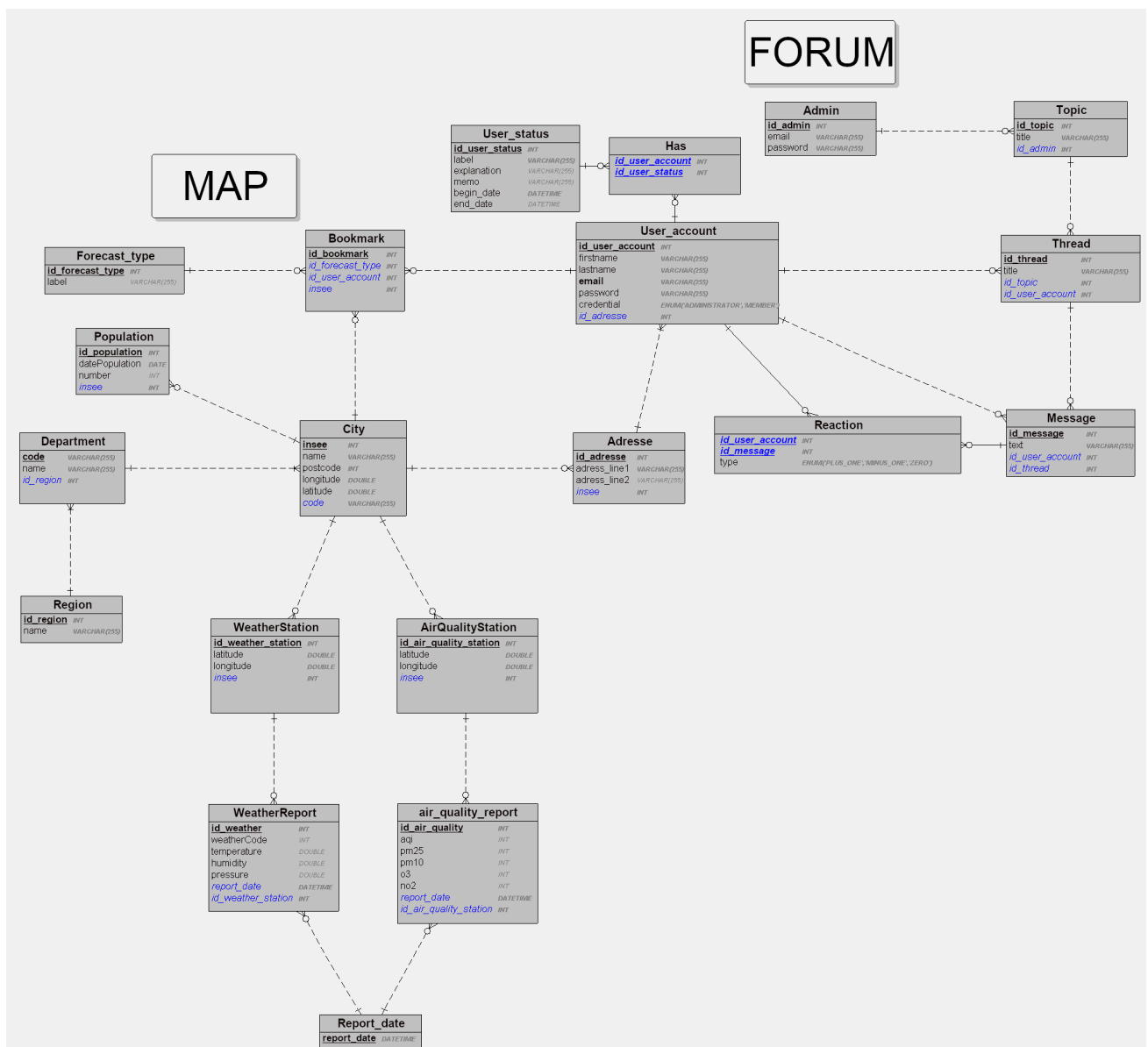
<u>Nom</u>	<u>Version</u>	<u>Référence</u>
MariaDB	15.1	https://mariadb.org/

4 FOCUS TECHNIQUE

4.1 Diagramme de classes métier



4.2 Modèle physique de données



4.3 Règles de développement coté back

Règles de développement

La documentation par Javadoc comprendra :

- Les *modules* – si le projet est modulaire – comprenant les noms d'auteur et le nom du module ;
- Les *packages* avec description de la fonction dans le projet ;
- Les *classes* avec description sur la signature puis les attributs, les méthodes et les classes internes ;

La documentation complémentaire comprendra :

- Les fichiers de configuration *pom.xml* et *persistence.xml* par leurs éléments dédiés aux descriptions ou via le commentaire en XML.

- Le fichier *README.md* résumant le projet dans son ensemble en *Markdown*.

Découpage en couches : à préciser quand vous aurez vu Spring Boot.

Découpage en couches :

décrire les différentes couches avec contrôleurs, DAO, classes techniques, DTO, etc.

Découpage en packages : organisation du code

Règles de nommage :

- Les DAOS ou Repositories ?
- Les DTOs ?
- Les classes de services ?
- *Les classes utilitaires ? exemple : classe DateUtils pour le traitement des dates*

« Ce point sera détaillé prochainement. »

5 TESTS ET INTÉGRATION

5.1 *Stratégie de tests*

Il y aura des tests unitaires à développer pour tester le code de toutes les classes de services.
S'aligner aux conventions de JUnit 5.

5.2 *Indicateurs de qualité de code*

Renseigner à 100% la Javadoc.

Règles de nommage des classes, des packages : conventions standards à respecter

Nom du garant de la qualité :