YouCode
POWERED BY SIMPLON

SIMPLON
.CO

Cahier des Charges – TonTin

- Garantir l’équité et éviter les répétitions

Page de Garde:

- Nom de l’application: TonTin
- Rotation Management Platform (au centre)
- Titre: Cahier des Charges
- Réalisé par: ismail baoud
- Encadré par: latifa amougay

Chapitre I:

Contexte du Projet

1. Introduction TonTin est une plateforme web conçue pour gérer des systèmes de rotation de groupes, inspirés du concept marocain de Dâr, sans inclure la gestion d’argent.
2. Problématique Les groupes traditionnels de rotation manquent souvent de transparence et de traçabilité. Il est difficile de suivre les tours, gérer les membres et assurer l’équité.
3. Solution Proposée
    - Organisation claire des membres
    - Gestion automatisée des rotations
    - Communication interne transparente
    - Assignation équitable des tours
4. Objectifs
    - Digitaliser la gestion des groupes par rotation
    - Assurer une visibilité complète des cycles
    - Améliorer la coordination et la communication

1.d Gestion des Rotations - Génération automatique - Édition manuelle pour exceptions - Rotation non répétitive garantie - Suivi des tours passés, en cours et futurs

1.e Communication - Chat de groupe - Annonces organisateur - Notifications: demandes, début de cycle, tours, état du groupe

1.f Tableau de Bord - Vue d'ensemble du groupe - Liste des membres - Calendrier des rotations - Suivi de l'activité et progression des cycles

2. Description Technique

Backend (Spring Boot 3) - Spring Security JWT - JPA / Hibernate + PostgreSQL - Architecture en couches: Controller / Service / DTO / Repository - Gestion centralisée des exceptions - Autorisation par rôle au niveau du groupe

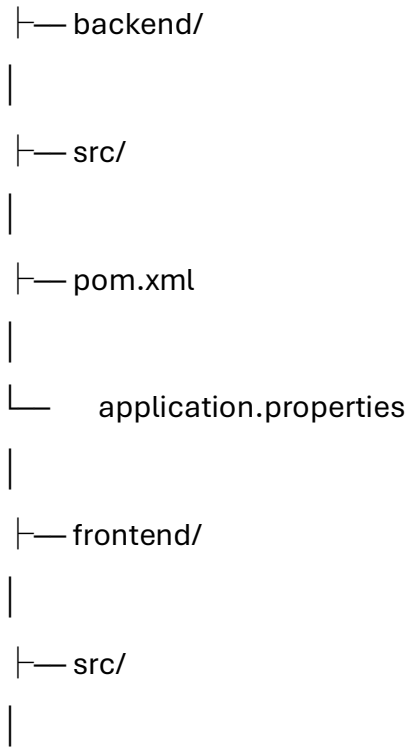
Frontend (Angular 17) - Angular Material / PrimeNG - RxJS pour la gestion d'état - Architecture modulaire et composants réutilisables - Interface responsive

Base de données - Tables principales: Users, Groups, GroupMembers, Rotations, Messages, Notifications - Permissions définies par groupe, pas globales

Structure du projet

TonTin/

platform/



```
├── angular.json
│
└── package.json
│
└── README.md
```

Installation & Setup - Backend: mvn clean install → mvn spring-boot:run (port 8080) - Frontend: npm install → ng serve (port 4200)

Tests - Backend: JUnit 5, Mockito (mvn test) - Frontend: Jasmine, Karma (ng test)

Déploiement - Docker, Railway / Render / Heroku, VPS Nginx, Spring Boot JAR

API - Swagger UI: /swagger-ui.html

Sécurité - JWT, validation des inputs, bonnes pratiques OWASP - Pas de traitement financier

Organisation et Modélisation - Planification JIRA: lien public à créer - Diagrammes Lucidchart: diagramme de classes + use cases (lien public)

Documentation et Présentation - Canva pour synthèse visuelle, lien public à créer