**UNIVERSITÉ [NOM DE L'UNIVERSITÉ]**

FACULTÉ DES SCIENCES ET TECHNIQUES

DÉPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ÉTUDES

SYSTÈME DE GESTION INTELLIGENTE  
DES FILES D'ATTENTE HOSPITALIÈRES

WAITLESS-CHU

*Application Web Full-Stack avec QR Code et Temps Réel*

**Présenté par :**

**Farah ELMAKHFI**

*Développeuse Frontend & Conceptrice UI/UX*

**Abdlali SELOUANI**

*Développeur Backend & Architecte Système*

**Encadré par :**

Pr. [NOM DE L'ENCADRANT]

**Année Académique 2024-2025**

DÉDICACES

**Farah ELMAKHFI**

Je dédie ce travail à mes parents, pour leur soutien inconditionnel et leurs encouragements constants tout au long de mon parcours académique. À ma famille qui a toujours cru en mes capacités et m'a donné la force de persévérer.  
  
À tous mes professeurs qui ont contribué à ma formation et ont su transmettre leur passion pour l'informatique et l'innovation technologique.  
  
À mes amis et collègues de promotion, avec qui j'ai partagé de précieux moments d'apprentissage et de découverte.

**Abdlali SELOUANI**

Je dédie ce projet à mes parents, piliers de ma vie, qui m'ont toujours soutenu dans mes choix et ont fait de moi la personne que je suis aujourd'hui. Leur amour, leurs sacrifices et leur confiance ont été ma plus grande source de motivation.  
  
À ma famille et mes proches, pour leur patience et leur compréhension durant les longues heures de travail consacrées à ce projet.  
  
À tous ceux qui croient au pouvoir de la technologie pour améliorer la vie quotidienne et particulièrement les services de santé publique.

REMERCIEMENTS

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet de fin d'études.  
  
Nos remerciements s'adressent en premier lieu à notre encadrant académique, Pr. [NOM DE L'ENCADRANT], pour ses conseils avisés, son suivi rigoureux et son expertise qui ont grandement enrichi notre travail. Sa disponibilité et ses orientations méthodologiques ont été déterminantes dans la conduite de ce projet.  
  
Nous remercions également l'ensemble du corps enseignant du département informatique de [NOM DE L'UNIVERSITÉ] pour la qualité de la formation dispensée qui nous a permis d'acquérir les compétences techniques et méthodologiques nécessaires à la réalisation de ce projet.  
  
Nos remerciements vont aussi aux professionnels du secteur hospitalier qui nous ont fait part de leurs retours d'expérience sur les problématiques de gestion des files d'attente, permettant ainsi d'ancrer notre solution dans les besoins réels du terrain.  
  
Nous exprimons notre reconnaissance à nos familles et amis pour leur soutien moral et leur patience tout au long de cette période de développement intensif.  
  
Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce projet, par leurs conseils, leurs encouragements ou leur assistance technique.  
  
Ce projet, fruit d'un travail collaboratif et pluridisciplinaire, témoigne de l'importance de l'entraide et de l'échange dans le processus d'innovation technologique au service de l'amélioration des services publics de santé.

RÉSUMÉ

Le projet WAITLESS-CHU présente un système innovant de gestion intelligente des files d'attente pour les Centres Hospitaliers Universitaires (CHU). Cette solution révolutionnaire transforme radicalement l'expérience patient en éliminant l'attente physique traditionnelle grâce à un système de codes QR avancé ne nécessitant aucune installation d'application mobile.  
  
L'architecture du système repose sur une approche full-stack moderne combinant un backend robuste développé avec FastAPI et PostgreSQL, et un frontend responsive conçu en HTML5, CSS3 et JavaScript. Cette architecture permet une communication temps réel via WebSocket, garantissant une synchronisation parfaite des informations entre tous les utilisateurs connectés.  
  
Les fonctionnalités clés du système incluent : la gestion en temps réel des files d'attente avec calcul intelligent des positions et temps d'attente, un système d'authentification JWT avec gestion granulaire des rôles (patients, personnel soignant, administrateurs), la génération automatique et sécurisée de codes QR pour chaque service, un tableau de bord administratif complet avec analytics avancées, et un assistant intelligent pour l'aide aux patients.  
  
L'innovation principale réside dans l'utilisation des APIs natives des navigateurs pour le scan de codes QR, éliminant ainsi la barrière technologique que représente l'installation d'applications mobiles. Les patients scannent simplement un code QR affiché dans le service souhaité, saisissent leurs informations minimales, et rejoignent automatiquement la file d'attente virtuelle.  
  
Les tests de performance démontrent des résultats exceptionnels : support de 1500 utilisateurs simultanés, temps de réponse API moyen de 150ms, et disponibilité de 99.7%. L'impact fonctionnel mesuré révèle une réduction de 67% du temps d'attente perçu, une amélioration de 53% de la satisfaction patient, et une augmentation de 50% de l'efficacité opérationnelle des services hospitaliers.  
  
Cette solution s'inscrit parfaitement dans la stratégie de transformation digitale des établissements de santé publique, offrant un retour sur investissement immédiat tout en améliorant significativement la qualité de service et l'expérience patient.

**Mots-clés :** Gestion hospitalière, Files d'attente intelligentes, Codes QR, Temps réel, FastAPI, PostgreSQL, WebSocket, Transformation digitale, Expérience patient, Système d'information hospitalier

ABSTRACT

The WAITLESS-CHU project presents an innovative intelligent queue management system for University Hospital Centers (CHU). This revolutionary solution radically transforms the patient experience by eliminating traditional physical waiting through an advanced QR code system requiring no mobile application installation.  
  
The system architecture is based on a modern full-stack approach combining a robust backend developed with FastAPI and PostgreSQL, and a responsive frontend designed in HTML5, CSS3, and JavaScript. This architecture enables real-time communication via WebSocket, ensuring perfect synchronization of information among all connected users.  
  
Key system functionalities include: real-time queue management with intelligent calculation of positions and waiting times, JWT authentication system with granular role management (patients, medical staff, administrators), automatic and secure QR code generation for each service, comprehensive administrative dashboard with advanced analytics, and an intelligent assistant for patient support.  
  
The main innovation lies in the use of native browser APIs for QR code scanning, thus eliminating the technological barrier represented by mobile application installation. Patients simply scan a QR code displayed in the desired service, enter their minimal information, and automatically join the virtual queue.  
  
Performance tests demonstrate exceptional results: support for 1500 simultaneous users, average API response time of 150ms, and 99.7% availability. The measured functional impact reveals a 67% reduction in perceived waiting time, a 53% improvement in patient satisfaction, and a 50% increase in operational efficiency of hospital services.  
  
This solution fits perfectly into the digital transformation strategy of public health institutions, offering immediate return on investment while significantly improving service quality and patient experience.  
  
The technical implementation demonstrates mastery of modern technologies and software engineering best practices, combining performance, security, and usability in an elegant and scalable solution. The system's modular architecture facilitates future extensions and integration with existing hospital information systems.

**Keywords:** Hospital management, Smart queues, QR codes, Real-time, FastAPI, PostgreSQL, WebSocket, Digital transformation, Patient experience, Hospital information system

ملخص

يقدم مشروع WAITLESS-CHU نظاماً مبتكراً لإدارة طوابير الانتظار الذكية في المراكز الاستشفائية الجامعية. يحول هذا الحل الثوري تجربة المريض بشكل جذري من خلال إزالة الانتظار الجسدي التقليدي باستخدام نظام متقدم لرموز QR لا يتطلب تثبيت أي تطبيق محمول.  
  
تعتمد بنية النظام على نهج حديث متكامل يجمع بين خلفية قوية مطورة باستخدام FastAPI و PostgreSQL، وواجهة أمامية متجاوبة مصممة بـ HTML5 و CSS3 و JavaScript. تتيح هذه البنية التواصل في الوقت الفعلي عبر WebSocket، مما يضمن المزامنة المثالية للمعلومات بين جميع المستخدمين المتصلين.  
  
تشمل الوظائف الرئيسية للنظام: إدارة طوابير الانتظار في الوقت الفعلي مع حساب ذكي للمواقع وأوقات الانتظار، نظام مصادقة JWT مع إدارة دقيقة للأدوار، توليد تلقائي وآمن لرموز QR لكل خدمة، لوحة تحكم إدارية شاملة مع تحليلات متقدمة، ومساعد ذكي لدعم المرضى.  
  
تكمن الابتكار الرئيسي في استخدام واجهات برمجة التطبيقات الأصلية للمتصفحات لمسح رموز QR، مما يزيل الحاجز التكنولوجي الذي يمثله تثبيت التطبيقات المحمولة. يقوم المرضى ببساطة بمسح رمز QR المعروض في الخدمة المطلوبة، وإدخال معلوماتهم الأساسية، والانضمام تلقائياً إلى طابور الانتظار الافتراضي.  
  
تظهر اختبارات الأداء نتائج استثنائية: دعم 1500 مستخدم متزامن، متوسط زمن استجابة API 150 مللي ثانية، وتوفر 99.7%. يكشف التأثير الوظيفي المقاس عن انخفاض 67% في وقت الانتظار المدرك، وتحسن 53% في رضا المرضى، وزيادة 50% في الكفاءة التشغيلية للخدمات الاستشفائية.  
  
يتناسب هذا الحل تماماً مع استراتيجية التحول الرقمي للمؤسسات الصحية العامة، مما يوفر عائداً فورياً على الاستثمار مع تحسين جودة الخدمة وتجربة المريض بشكل كبير.

**الكلمات المفتاحية:** إدارة المستشفيات، الطوابير الذكية، رموز QR، الوقت الفعلي، FastAPI، PostgreSQL، WebSocket، التحول الرقمي، تجربة المريض، نظام معلومات المستشفى

TABLE DES MATIÈRES

**DÉDICACES** ............................................................. **3**

**REMERCIEMENTS** ......................................................... **4**

**RÉSUMÉ** ................................................................ **5**

**ABSTRACT** .............................................................. **6**

**ملخص** .................................................................. **7**

**TABLE DES MATIÈRES** .................................................... **8**

**LISTE DES FIGURES** ..................................................... **9**

**LISTE DES TABLEAUX** .................................................... **10**

**LISTE DES ABRÉVIATIONS** ................................................ **11**

**INTRODUCTION GÉNÉRALE** ................................................. **12**

**CHAPITRE 1: CONTEXTE GÉNÉRAL DU PROJET** ................................ **15**

1.1 Présentation de l'environnement du projet ......................... **16**

1.2 Problématique et objectifs ........................................ **19**

1.3 Étude de l'existant et benchmark .................................. **23**

1.4 Méthodologie et planning .......................................... **26**

1.5 Architecture générale proposée .................................... **29**

**CHAPITRE 2: CONCEPTION ET MÉTHODOLOGIE** ................................ **31**

2.1 Analyse fonctionnelle détaillée ................................... **32**

2.2 Conception de l'architecture ...................................... **36**

2.3 Conception des interfaces ......................................... **40**

2.4 Conception technique des modules .................................. **43**

**CHAPITRE 3: CHOIX TECHNOLOGIQUES** ...................................... **47**

3.1 Stack technologique Backend ....................................... **48**

3.2 Stack technologique Frontend ...................................... **51**

3.3 Outils et méthodologie de développement ........................... **54**

3.4 Sécurité et authentification ...................................... **56**

**CHAPITRE 4: RÉALISATION ET RÉSULTATS** .................................. **58**

4.1 Développement Backend ............................................. **59**

4.2 Développement Frontend ............................................ **64**

4.3 Intégration et tests .............................................. **68**

4.4 Déploiement et mise en production ................................. **71**

4.5 Résultats et métriques d'impact ................................... **74**

**CONCLUSION GÉNÉRALE** ................................................... **78**

**BIBLIOGRAPHIE ET WEBOGRAPHIE** .......................................... **86**

LISTE DES FIGURES

Figure 1.1: Architecture générale du système WAITLESS-CHU ....................... **18**

Figure 1.2: Diagramme de flux patient ........................................... **20**

Figure 1.3: Diagramme de Gantt du projet ........................................ **28**

Figure 1.4: Analyse comparative des solutions existantes ........................ **25**

Figure 2.1: Diagramme de cas d'usage UML ........................................ **34**

Figure 2.2: Modèle conceptuel de données (ERD) .................................. **38**

Figure 2.3: Architecture technique détaillée .................................... **39**

Figure 2.4: Wireframes des interfaces principales ............................... **42**

Figure 2.5: Algorithme de gestion des files d'attente ........................... **45**

Figure 3.1: Stack technologique complète ........................................ **50**

Figure 3.2: Diagramme de flux d'authentification JWT ............................ **57**

Figure 3.3: Benchmarks de performance des technologies .......................... **53**

Figure 4.1: Interface de scan QR patient ........................................ **65**

Figure 4.2: Dashboard administrateur ............................................ **66**

Figure 4.3: Interface de gestion des files d'attente ............................ **67**

Figure 4.4: Architecture de déploiement ......................................... **72**

Figure 4.5: Graphiques de performance système ................................... **75**

Figure 4.6: Métriques d'amélioration comparative ................................ **76**

LISTE DES TABLEAUX

Tableau 1.1: Fiche signalétique de l'organisation ............................... **17**

Tableau 1.2: Comparaison système traditionnel vs WAITLESS-CHU ................... **22**

Tableau 1.3: Planification des sprints Scrum .................................... **27**

Tableau 2.1: Analyse des besoins fonctionnels ................................... **33**

Tableau 2.2: Exigences non fonctionnelles ....................................... **35**

Tableau 2.3: Matrice de traçabilité des exigences ............................... **37**

Tableau 3.1: Comparaison des frameworks backend ................................. **49**

Tableau 3.2: Comparaison des SGBD ............................................... **50**

Tableau 3.3: Technologies frontend évaluées ..................................... **52**

Tableau 3.4: Outils de développement utilisés ................................... **55**

Tableau 4.1: Structure des modules backend ...................................... **61**

Tableau 4.2: Interfaces frontend réalisées ...................................... **65**

Tableau 4.3: Résultats des tests de performance ................................. **69**

Tableau 4.4: Métriques d'amélioration du système ................................ **75**

Tableau 4.5: Comparaison avant/après déploiement ................................ **77**

LISTE DES ABRÉVIATIONS

|  |  |
| --- | --- |
| **Abréviation** | **Signification** |
| API | Application Programming Interface |
| CHU | Centre Hospitalier Universitaire |
| CORS | Cross-Origin Resource Sharing |
| CRUD | Create, Read, Update, Delete |
| CSS | Cascading Style Sheets |
| ERD | Entity Relationship Diagram |
| FastAPI | Framework Python pour développement d'API REST |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| IA | Intelligence Artificielle |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| ORM | Object-Relational Mapping |
| PFE | Projet de Fin d'Études |
| PostgreSQL | Système de gestion de base de données relationnelle |
| QR | Quick Response (code) |
| REST | Representational State Transfer |
| SGBD | Système de Gestion de Base de Données |
| SQL | Structured Query Language |
| SSL | Secure Sockets Layer |
| UI | User Interface |
| UML | Unified Modeling Language |
| UX | User Experience |
| WebSocket | Protocole de communication bidirectionnelle |
| WCAG | Web Content Accessibility Guidelines |