

Cahier de charges
Factory de mots de passes

Alaa Abdelbaki
Youssef Derouiche

Table de matières

1	Introduction	2
1.1	Contexte	2
1.2	Objectifs du projet	2
1.3	Portée	2
2	Périmètre du projet	3
2.1	Fonctionnalités attendues	3
2.2	Technologies utilisées	3
3	Public cible	4
3.1	Profil des utilisateurs	4
3.2	Nombre de personnes pour les tests	4
4	Scénario des tests utilisateurs	5
4.1	Objectifs des tests utilisateurs	5
4.2	Méthodologie des tests	5
4.3	Scénarios de tests	5
5	Critères d'évaluation et de réussite	6
5.1	Indicateurs de performance - KPI	6
6	Contraintes et risques	7
6.1	Contraintes techniques	7

1 Introduction

1.1 Contexte

Le projet **"Factory de mots de passes"** s'inscrit dans le contexte de l'apprentissage automatique pour la génération des mots de passes. Il consiste à développer une application qui implémente un réseau de neurones avec des options permettant d'entraîner le modèle sur des données d'entrée et l'évaluer en générant des nouvelles séquences à partir de quelques caractères de départ entrés par l'utilisateur. Cette solution permet ainsi de générer automatiquement des mots de passes qui respectent les conventions d'une entreprise ou d'un domaine spécifique.

1.2 Objectifs du projet

Ce projet a les objectifs suivants à atteindre:

- Développer un modèle intelligent en utilisant la bibliothèque PyTorch [2] de Python [1].
- Développer des fonctions d'entraînement et d'évaluation pour le modèle.
- Préparer un corpus d'entraînement contenant des mots de passes respectant une convention.
- Développer une solution logicielle pour mieux utiliser la solution intelligente.

1.3 Portée

Cette application sera destinée pour un grand public qui veut générer des mots de passes facile à mémoriser en entrant un mot clé.

2 Périmètre du projet

2.1 Fonctionnalités attendues

La solution proposée doit être capable de fournir plusieurs fonctionnalités aux utilisateurs tels que:

- Temps d'entraînement minimal.
- Générer des mots de passes selon des critères bien déterminés.
- Modifier les données d'apprentissage pour mettre à jour les politiques de génération des mots de passes si nécessaire.
- Utilisation simple et non complexe à l'aide d'une interface graphique facile à utiliser.

2.2 Technologies utilisées

Afin de pouvoir élaborer ce projet, on a eu recours aux différents langages de programmation, frameworks et outils techniques tels que:

- **Python:** Pour le développement du réseau de neurones et d'un serveur qui permettra la communication entre une application et/ou un site web avec le modèle entraîné.
- **Flutter:** Pour le développement d'une application mobile/web qui sera utilisée par l'utilisateur pour utiliser le modèle de génération de mots de passes.

3 Public cible

3.1 Profil des utilisateurs

La solution est destinée à:

- Administrateurs système au sein d'une entreprise.
- Utilisateurs d'applications nécessitant une création de compte.
- Personnes ayant des compétences faibles à moyennes en informatique.

3.2 Nombre de personnes pour les tests

- **Phase 1:** 10 utilisateurs (Phase exploratoire)
- **Phase 2:** 50 utilisateurs (Test grandeur nature)

4 Scénario des tests utilisateurs

4.1 Objectifs des tests utilisateurs

- Identifier les défaillances du système.
- Tester le temps d'apprentissage et d'exécution.
- Tester la robustesse des mots de passes générés.
- Corriger les problèmes identifiés.
- Améliorer les paramètres d'apprentissage pour améliorer le temps d'entraînement.

4.2 Méthodologie des tests

- Test avec un corpus de mots de passes différent du corpus d'entraînement.
- Tester avec une partie du corpus d'entraînement qui est dédiée au test.

4.3 Scénarios de tests

- Entraîner le modèle avec un corpus de mots de passes de taille importante dont chaque mot de passe valide des conditions prédéfinies.
- Modification des hyper paramètres du réseau de neuronne d'entraînement et comparer le résultat les autres qui utilisent des paramètres différents.

5 Critères d'évaluation et de réussite

5.1 Indicateurs de performance - KPI

Identifier les KPIs à suivre pour mesurer la réussite des tests.

Exemple:

- Taux de confiance important lors de l'entraînement ($> 90\%$)
- Test avec un corpus de test similaire au corpus d'entraînement et calculer le tax de réussite de prédiction des mots de passes.
- Temps nécessaire pour prédire un mot de passe.

6 Contraintes et risques

6.1 Contraintes techniques

- Temps d'entraînement important lié à la taille de corpus d'entraînement.
- Trouver un corpus d'entraînement qui couvre plusieurs exigences des mots de passes.

References

- [1] Fondation Python Software. Langage de programmation python. <https://www.python.org>, 2023. Consulté le 8 novembre 2024.
- [2] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: Une bibliothèque de deep learning performante et impérative. <https://pytorch.org>, 2019. Consulté le 8 novembre 2024.