40104/Secrets Manager

Requirements Engineering “Secrets Manager”

Task

Describe your project or any project neatly using the methods of Requirements Engineering.

Project Description

This is my base pet project of application for keep secrets. This small application should have a client connected to the backend and store the secrets encrypted in the database. In the process of completing this project, I would like to get acquainted with the Golang programming language.

Functional requirements

□ **User authentication**

- The system must have multi-factor authentication
- The system must support external authentication

□ **Creating folder**

- The system must support storing secrets in folders

□ **Creating secret**

- The system must support creating secrets and placing them in folders

□ **Share with users**

- The system must support sharing secrets with other users
- The system must support sharing folders with other users

□ **API**

- The system must support an API for creating custom web and desktop applications

□ **Plugin**

- The system must have automation plugins for all current browsers

□ **History of changes**

- The system must support change history functionality

□ **Roles**

- The system must support access separation by roles

□ **Secret generation**

- The system must be able to generate passwords, secrets, keys and certificates

□ **Reports**

- The system should generate a report on leaked secrets
- The system should do a self-health check and generate an error report

---

Non-functional requirements

□ **Security**

- The system must store data in encrypted form
- The system must send data in encrypted form

□ **Performance**

- The system must be highly efficient

□ **Reliability**

- The system must be reliable

□ **High availability**

- The system must support high availability mode
- The system must support external authentication

□ **Usability**

- The system should have a user-friendly interface and intuitive navigation

---

Transition requirements

□ **Data migration**

- The system must support data import
- The system must support data export

□ **Tested**

- The system must be tested for stability
- The system must be tested for security

□ **Team management**

- The system must support command operation

Interface requirements

□ **User Interface**

- The system should have a sticker design

□ **Multi languages support**

- The system must support many languages
