info@apriorit.com
www.apriorit.com

Case study

Developing a Custom Secrets Management Desktop Application for Secure Password Sharing and Storage

A US-based cybersecurity company hired us to create secrets management software for internal use. They were looking for a team with strong cybersecurity expertise and keen knowledge of desktop application development on Windows, macOS, and Linux.

The Apriorit team designed and developed a secrets manager as an on-demand solution, taking into consideration all of the client's requests.

The client

Our client is a US-based company that delivers high-end cybersecurity products and services to customers worldwide. The company offers top-notch protection against various cyber threats, including malware, data breaches, and phishing attacks.

The challenge

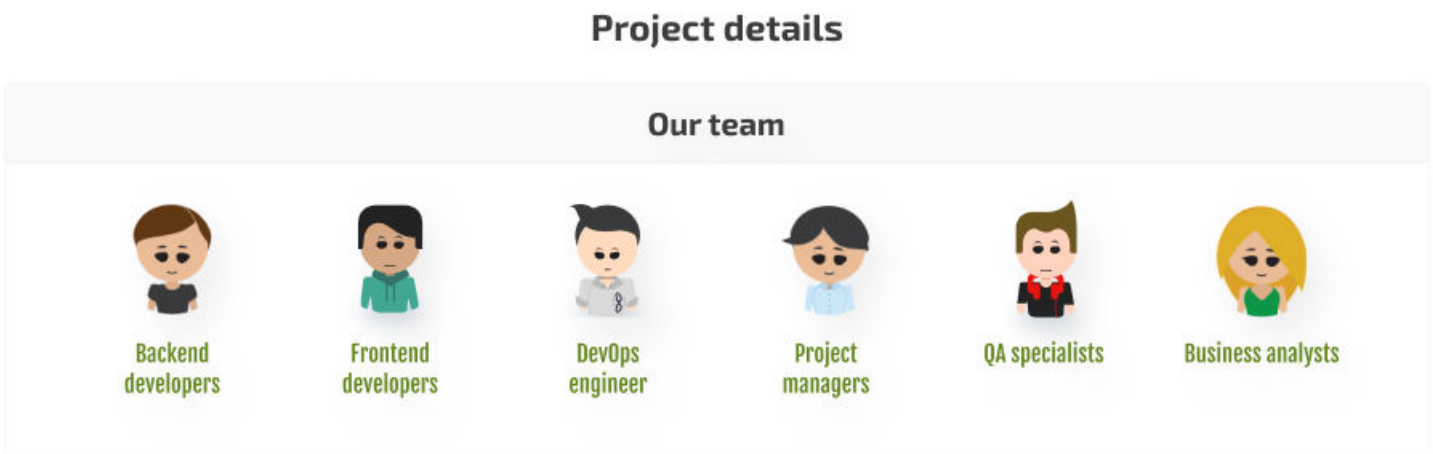
The client approached Apriorit with a request for developing a custom secrets management desktop application. Since they needed an application for inside use, the client had a prepared list of features to develop, including secrets sharing functionality, a secrets manager, and an administrator dashboard.

We also had to fulfill some additional requirements:Working on this project, we relied on the expertise of Apriorit professionals from several fields and the following technology stack:

- On-demand delivery
- Zero-knowledge architecture
- Compliance with SOC2, SOC3, C5, ISO 27001, and GDPR requirements
- Support for Windows, macOS, and Linux platforms

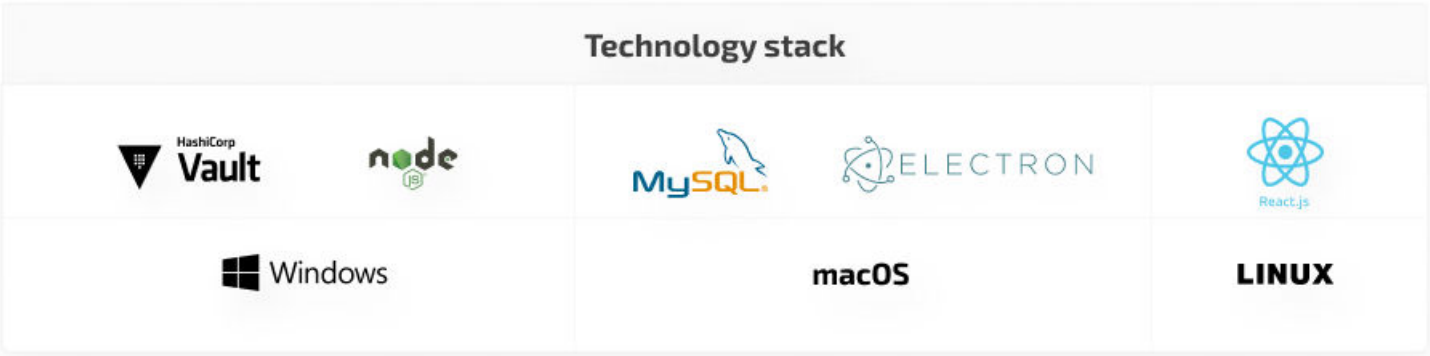
Our approach

After receiving the initial requirements, we analyzed the scope of work and assembled a team of experts with relevant expertise for developing a custom secrets management desktop app.



Project details

Technology stack



To establish clear and effective communication with our client, we used Slack and Jira for managing all project-related tasks. We regularly updated the client on the project's progress with weekly reports on the status of the secrets management solution development and held demos every two weeks to showcase the latest progress and gather feedback.

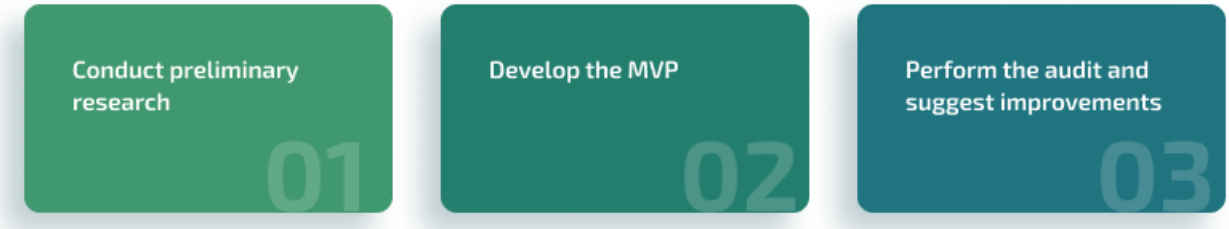
The result

Our team designed and implemented a custom secrets management software development process that satisfied the client's requirements and met the established project deadlines. As a result, our client received an effective tool for secrets management and data protection.

How we did it

To deliver the high-quality product and create a desktop app for secrets management within set deadlines, we thoroughly planned our project workflow and created a three-stage plan.

Project stages



Stage 1: Conduct preliminary research

The initial stage of our project started with conducting preliminary research so we could lay the groundwork for developing a solid product. At this stage, our business analysts gathered and analyzed the client's initial requirements, researched competitors, and helped our client and development team finalize the product vision for the minimum viable product (MVP).

Let's take a closer look at all actions taken during this stage.

Performing competitor research

Before we started to build a custom tool for secrets management, our business analysts investigated and analyzed the client's competition in the US market. As a result of this research, we created a comprehensive report and presented it to the client. Here are the key features of this report:

- A list of all relevant competitors in the industry
- An overview of all features that competitors offer
- A detailed analysis of the market size and future prospects
- A list of audits and security requirements declared by competitors

This research became a solid foundation for planning and developing the product strategy, providing a better understanding of the market and the opportunity to adjust our approach as needed.

Defining the scope of work for the MVP

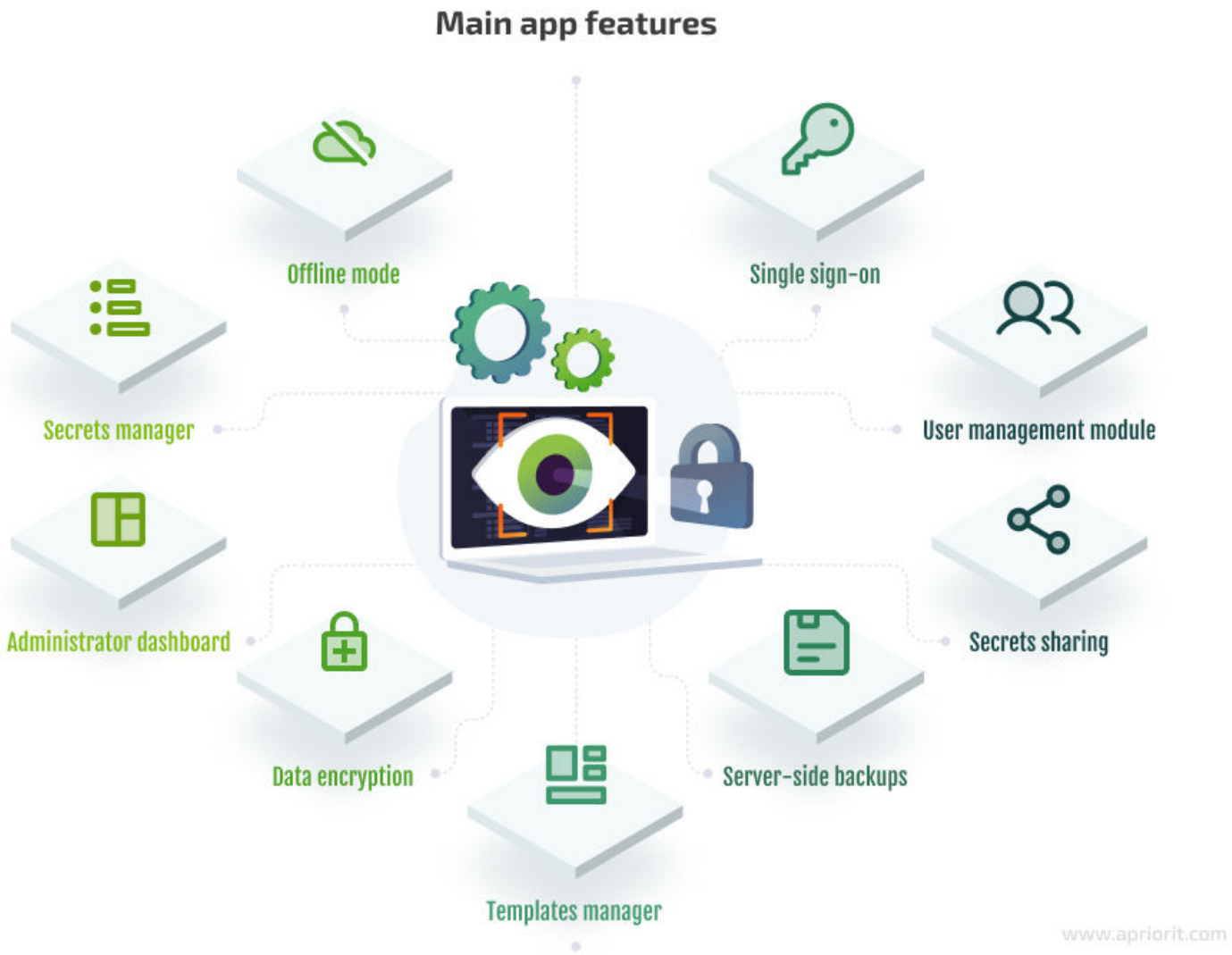
Using information gathered at the research stage, we created a detailed project estimate based on the following metrics:

- Required resources
- Scope of work
- Projected timeline

Our team approved these estimates with the client and moved on to the next stage.

Stage 2: Develop the MVP

Based on our research and the client's requirements, we started working on an MVP with the following features:



The Apriorit team of backend and frontend developers successfully delivered an MVP that included all of the requested features.

Let's look closer at the specifics of working with some of these features.

Implementing offline mode

One of our client's key requirements was support for offline mode in their desktop app. This would allow users to easily install the application and immediately start using the secrets manager, even without an internet connection.

The Apriorit team focused on improving the user experience and gathering valuable feedback from potential users. Offline mode features also allowed us to show demo sessions to potential users without any interruptions or connectivity issues.

Providing server-side backups and sharing features

We also introduced server-side backups and sharing features using HashiCorp Vault. This allowed for user authentication with the help of single sign-on (SSO). With this feature in place, users can seamlessly access their data across multiple devices.

To enhance the product's security, we implemented this feature using the following:

- 256-bit AES encryption for secrets data
- RSA keys for sharing data
- Data/Key Encryption Key (DEK, KEK) for communication with the server

Developing the administrator dashboard

Another fundamental requirement was an administrator dashboard that provides additional control and management features. With the help of this dashboard, administrators can perform several key functions, including:

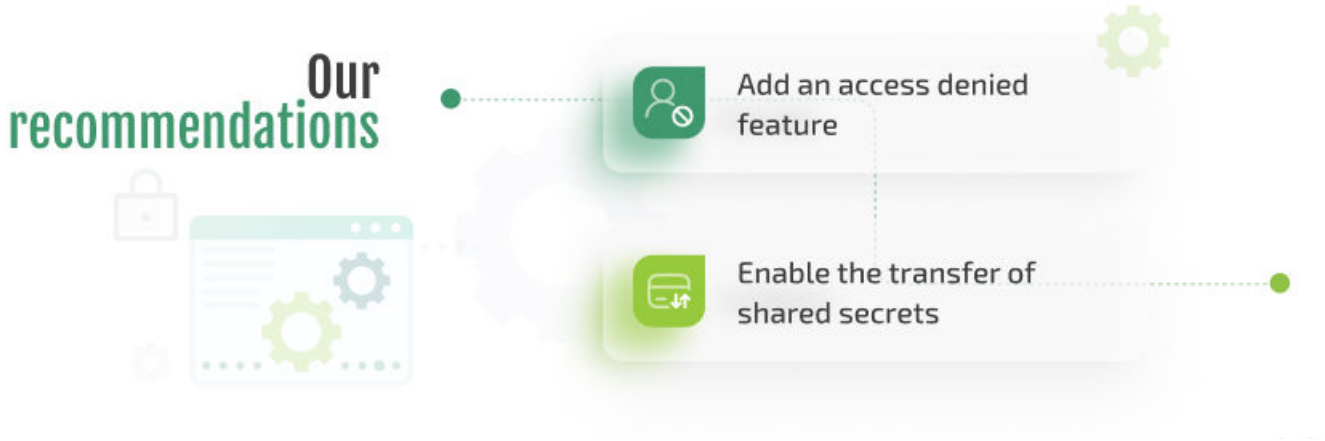
- Managing users
- Managing secrets templates
- Controlling the security score for each user
- Configuring customized reports and security alerts

Overall, the administrator dashboard provided our client with improved visibility and control over their user base, allowing for future improvements to the user experience.

Stage 3: Perform an audit and suggest improvements

After finishing with MVP development, we conducted a comprehensive security audit. Based on the results, we thought about ways to improve the security features of the client's application. This audit allowed us to identify hidden vulnerabilities and provided valuable insights into potential threats. By eliminating these threats, we were able to ensure top-level protection of client data. After conducting all audits, we offered to add two features to improve the security of the final product.

Based on the audit results and the client's feedback, we suggested implementing an access denied feature to improve the application's security. This feature would allow administrators to deny access to corporate secrets for specific users, adding an extra layer of protection for sensitive data.



Also, our developers added the ability for shared secrets to be transferred to a new owner. This allowed users to pass ownership of their secrets to others if needed. This is a useful feature when, say, a person is no longer employed and doesn't need access to corporate secrets, so an account can be further transferred to a new employee.

Challenges and solutions

Even though our team faced a few challenges during secrets manager desktop application development, we were able to find an effective solution to all issues thanks to open communication with our client and within our team.

The main challenges for our team were the following:

- **Choosing the right approach to implementing sharing mode**. While the client knew they wanted to have a secrets sharing feature in their application, there was no unified vision on how this feature would be implemented. We explored how competitors approach secrets sharing and what solutions were available on the market. We decided to use HashiCorp Vault, as it appeared to be the most effective solution.

- **Implementing a custom secrets vault**. While HashiCorp Vault can be used for storing secrets, it doesn't really support the zero-knowledge architecture requested by the client. Thus, we had to develop a custom secrets sharing algorithm to ensure proper secrets encryption while using HashiCorp Vault.

- **Adapting Electron to macOS and Linux.** The Electron framework works seamlessly with Windows. However, we found that some features required adaptation for Linux and macOS. To address this, we carried out extra adaptation and testing separately for each platform.

The impact

As a result of our partnership, the client received an efficient and competitive solution for safeguarding their sensitive information. Deploying the product delivered by the Apriorit team helped the client to improve their security posture and protect their valuable assets.

Based on client feedback, the overall security score of the company improved by 30%. Apriorit developers provided our client with all necessary tools for effectively managing their secrets and making sure that only authorized and approved users had access to critical data within the company.

Want to improve your application's security score? Contact Apriorit so we can start developing an effective security solution together!