



Developing the GUI for a global company's memory analysis tool

Industry:

Enterprise

The Client:

Our client is a well-established hardware-based security solutions provider. They provide comprehensive Hardware-Based Security Solutions for Protecting PCs, Network and Consumer Devices. These hardware-assisted services include software protection and anti-tamper, malware protection, information privacy, and digital media rights management. The technology can be applied in computer systems, set top boxes, network, and consumer devices.

Business objective:

Our client planned to develop a memory analysis tool (forensic tool). The purpose of the project was developing an application GUI that would work in conjunction with the client PCIe card and kernel software to provide information and visualize physical memory, virtual memory, memory binary files and present automatically detected malicious activity and modifications in memory. All data structures and text sections in the OS kernel, processes, drivers, registry etc should be accessed and presented.

Solution:

ennovate was required to develop a GUI for a software which would communicate with the low level API developed by the client.

We followed agile development methodology and prototyping method during the software development life cycle. The software is multi-tier application which was divided in mainly three layers. The **UI Layer** represents the data to the user. The **Logic Layer** communicates with the low level API layer to get the data. And the **Low Level API**, interacts with OS API and hardware drivers to provide the data. We developed and designed the GUI for the software and making it functional using low level API to be developed by client. Last step involved the integration of UI with the low level AP.

Challenge:

Low level API was planned to be developed in parallel with the UI development. Owing to a delay in development of low level API, ennovate team started developing mock implementation of the low level API.

The basic purpose of the software is to analyze the memory of the device. We needed to show and keep in memory only that much data which is visible on the screen. We took this challenge and developed customized components which were fetching and loading the data on demand and flushing it as soon as it goes out of view.

It helped us a lot in saving the memory and meeting the constraint of the project. Almost all the objects in the software are complex data structure. The challenge was to represent this complex and bulky data to users in an accessible and comprehensible manner. We divided and organized the data to be displayed in logical manner which provides better user experience.

Technology:

- Java Technology was used to develop the software to achieve platform independency
- Supporting multiple Operating Systems.
- GUI was developed using Java Swing
- Java Native Interface was used for communicating with the low level API to be developed by the client.

Benefits:

- **Multiple options for GUI** for various functionalities, thus enabling the client to make a decision to chose the best possible UI.
- Adopted **Mock implementation approach** due to unavailability of low level API modules
- Project **timelines were maintained** and time to market was reached.

ennovate
Technologies

www.ennovatetech.com

Case Study