# Implementation

## Front end implementation

As the interface designs were created, it provided a clear structure of what the application should visually look like, and implementing these designs were also very easy, this was due to the use of graphical class library known as “Windows Forms”, which provided the necessary components known as “controls” needed to adhere to the standards of the designs.

Below shows the implemented “Overview” section:

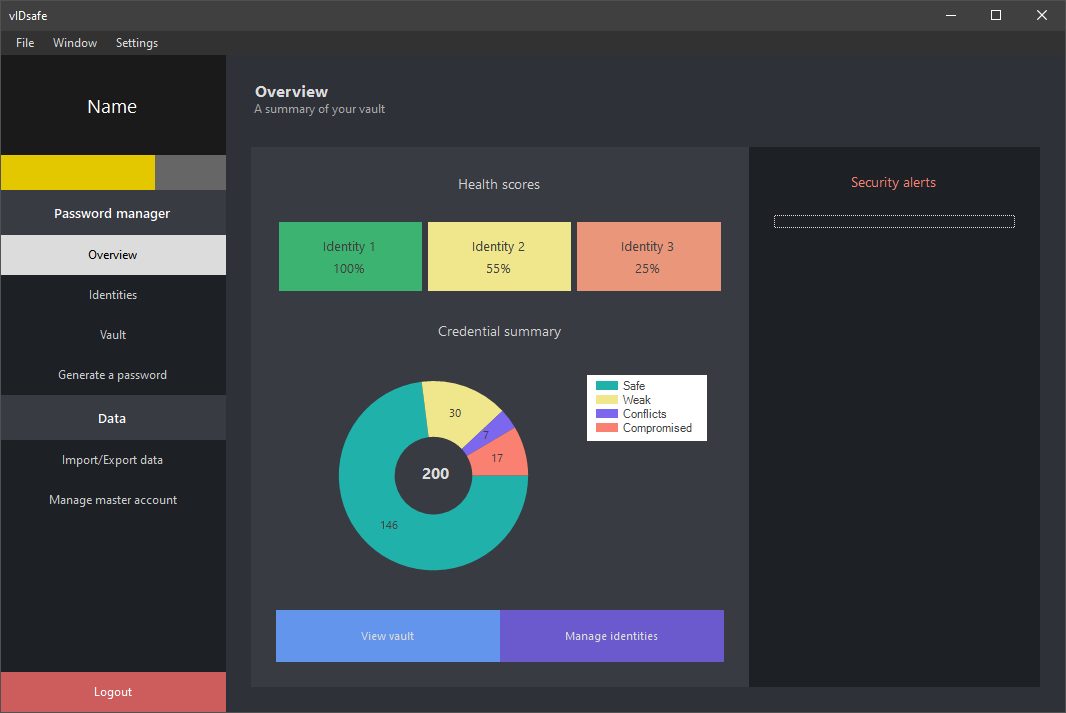


Figure 5‑1 Implemented overview section

Much like the designed screen in Figure 4‑6, above shows a slightly modified version in which the layout is identical although the colour scheme was changed for consistency. When looking further into this screen, it’s also noticeable that just about everything is placed into a panel, which is a control given by WinForms, the reason for this was because it helps organise the layout of the screen through grouping its sub controls as well as aligning and making it more responsive through the use of properties such as “docking” and “anchoring” as well as “margin” and “padding”; the title panel for an example is docked at the very top meaning regardless of the screen size, it will always be anchored to the top. The inner panel showing the “health scores” and the “credential summary” as well as the “security alerts” uses a padding of 25 for all directions to provide a 25-pixel gap to separate them from the rest of the components in the screen. Something else that’s also noticeable is the design of the buttons; these all use the “flat” style as their appearance which diverts from the average windows button which may have borders, which wouldn’t compliment the overall design of the screen.

As previously said, just about everything is inside of a panel, and this includes the different sections of the application as well, which are all part of a “navigation” form/screen, this is shown below:

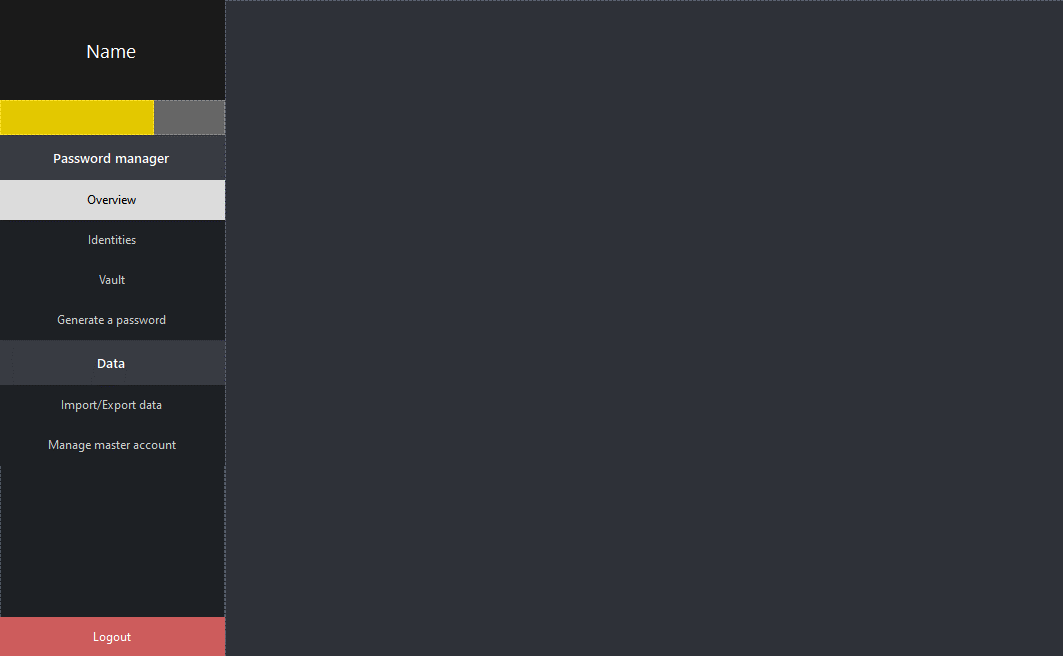


Figure 5‑2 Navigation screen

Each section is split into a form of its own, and when navigated to it by one of the buttons on the navigation panel on the left, it simply opens that form onto the panel on the right inside of this navigation form, this is known as a child form and it was made possible through code in which the child form based on the navigation button is added as a control inside of the panel. The reason why it was done this way was so that it wouldn’t be necessary to then duplicate the navigation panel onto each form which would create redundancy as it’d also mean that the navigation code would be duplicated for each. This also meant that a lot of memory would be saved considering these child forms are closed before opening another, this was also done in code by assigning the current child form as the “active” form.

Regarding the sections/screen of the application, one in particular had to be altered separately, this was the “Application settings” section which was converted into a toolbar:

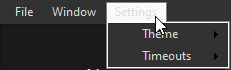


Figure 5‑3 Settings toolbar

The reason for this change was because it felt unnecessary to have it as a part of the navigation panel as it seemed “out of place” due to it having fewer content within, hence why the “ToolStrip” was used as a solution to make the options as simple and easily accessible.

## Back end implementation